

# TURNOVER<sup>®</sup> Lifecycle Manager

Administrator's Guide

Release 2.100

## Publication information

TN4A-2100-02 (November 2022)

Information in this publication is subject to change. Changes will be published in new editions or technical newsletters.

## Copyright notice

TURNOVER® Lifecycle Manager (the Programs and associated materials) is a proprietary product of SoftLanding Systems, Inc. – a division of UNICOM Global. The Programs have been provided pursuant to License Agreement containing restrictions on their use. The programs and associated materials contain valuable trade secrets and proprietary information of SoftLanding Systems, Inc. and are protected by United States Federal and non-United States copyright laws. The Programs and associated materials may not be reproduced, copied, changed, stored, disclosed to third parties, and distributed in any form or media (including but not limited to copies on magnetic media) without the express prior written permission of SoftLanding Systems, Inc., 1 Vose Farm Road, Suite 110, Peterborough, NH 03458, USA.

TURNOVER® Lifecycle Manager (the Programs and associated materials) is a proprietary product of UNICOM Systems, Inc. – a division of UNICOM Global. The Programs have been provided pursuant to License Agreement containing restrictions on their use. The programs and associated materials contain valuable trade secrets and proprietary information of UNICOM Systems, Inc. and are protected by United States Federal and non-United States copyright laws. The Programs and associated materials may not be reproduced, copied, changed, stored, disclosed to third parties, and distributed in any form or media (including but not limited to copies on magnetic media) without the express prior written permission of UNICOM Systems, Inc., UNICOM Plaza Suite 310, 15535 San Fernando Mission Blvd., Mission Hills, CA 91345 USA.

## TURNOVER® Lifecycle Manager

© Copyright 2005-2023 All Rights Reserved. Softlanding Inc. – a division of UNICOM Global.

No part of this Program may be reproduced in any form or by electronic means, including the use of information storage and retrieval systems, without the express prior written consent and authorization of SoftLanding Systems, Inc.

No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, without prior written permission from SoftLanding Systems, Inc.

© Copyright 2005-2023 All Rights Reserved. UNICOM Systems, Inc. – a division of UNICOM Global.

No part of this Program may be reproduced in any form or by electronic means, including the use of information storage and retrieval systems, without the express prior written consent and authorization of UNICOM Systems, Inc.

No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, without prior written permission from UNICOM Systems, Inc.

## Disclaimer

We cannot guarantee freedom from, or assume any responsibility or liability for technical inaccuracies or typographical errors. The information herein is furnished for informational use only and should not be construed as a commitment by SoftLanding Systems, Inc. – a division of UNICOM Global.

## Trademarks

The following are trademarks or registered trademarks of Macro 4 Limited, SoftLanding Systems, Inc. and/or of UNICOM Systems, Inc. in the United States and/or other jurisdictions worldwide: UNICOM, UNICOM Systems, Macro 4, SoftLanding, AUTOMON, CENTRAL, EXPRESSDESK, SUPERMON, TURNOVER.

For a list of third-party products, companies, trademarks, and registered trademarks that might be referenced in this manual, see [www.softlanding.com/trademarks](http://www.softlanding.com/trademarks).

# Contents

|          |                                                                |           |
|----------|----------------------------------------------------------------|-----------|
| <b>1</b> | <b>TURNOVER® Lifecycle Manager Administrator's Guide</b>       | <b>8</b>  |
|          | Choosing the TURNOVER® Lifecycle Manager administrator         | 8         |
| <b>2</b> | <b>Application planning</b>                                    | <b>10</b> |
|          | Overview of TURNOVER® Lifecycle Manager                        | 10        |
|          | How TURNOVER® Lifecycle Manager improves productivity          | 12        |
|          | Setting up TURNOVER® Lifecycle Manager                         | 14        |
|          | Security                                                       | 14        |
|          | About library structure                                        | 15        |
|          | Establishing source archive policies                           | 15        |
|          | Where do TURNOVER® Lifecycle Manager projects come in?         | 17        |
|          | Designing your TURNOVER® Lifecycle Manager change environments | 17        |
|          | Modeling with application definition models                    | 19        |
|          | Application definitions                                        | 32        |
|          | About application diagrams                                     | 33        |
|          | Workflow                                                       | 35        |
|          | Creating application definitions                               | 37        |
|          | Copying an application model                                   | 37        |
|          | Using the New Application Definition wizard                    | 39        |
|          | Introduction to the Application Designer                       | 44        |
| <b>3</b> | <b>Configuring auditing needs</b>                              | <b>48</b> |
|          | Define a software development process                          | 48        |
|          | Audit a change                                                 | 50        |
| <b>4</b> | <b>Change process</b>                                          | <b>52</b> |
|          | Form promotion processing flow                                 | 53        |
|          | Source-only objects                                            | 58        |
|          | Source-compiled objects                                        | 59        |
|          | Non-source objects                                             | 60        |
|          | Additional rules and conditions                                | 60        |
|          | New logical files                                              | 64        |
|          | Pre-run and post-run commands                                  | 66        |
|          | Pre-run and post-run command scenarios                         | 68        |
|          | Deleting objects                                               | 68        |

|                                                  |            |
|--------------------------------------------------|------------|
| Form copy and object delete rules                | 68         |
| Delete test object rule                          | 70         |
| <b>5 Source Stream Files</b>                     | <b>72</b>  |
| Introduction to source stream files              | 72         |
| Application Definitions                          | 80         |
| <b>6 Managing object types, schemes and data</b> | <b>86</b>  |
| Creating a CM scheme                             | 87         |
| Overriding CM methods                            | 89         |
| Changing CM methods                              | 90         |
| Create CM method programs                        | 96         |
| Create type codes to use a CM scheme             | 107        |
| Using CM schemes                                 | 107        |
| Example change management schemes                | 111        |
| <b>7 Using create commands</b>                   | <b>131</b> |
| Compiling and creating the object                | 131        |
| Changing the object after promotion              | 132        |
| <b>8 Ownership, profiles and jobs</b>            | <b>134</b> |
| Object ownership                                 | 134        |
| User profiles                                    | 135        |
| Autostart jobs                                   | 139        |
| <b>9 Security recommendations</b>                | <b>149</b> |
| Object owners                                    | 149        |
| Production libraries                             | 150        |
| Setting specific object authority                | 151        |
| Checking your system for security exposure       | 154        |
| <b>10 Emergency changes</b>                      | <b>158</b> |
| Using an emergency change application            | 158        |
| Using the multiple checkout feature              | 159        |
| Resolving concurrent changes                     | 159        |
| <b>11 Managing multiple versions</b>             | <b>161</b> |
| Related application change scenarios             | 163        |
| Setting up application relationships             | 165        |
| Using related applications                       | 166        |
| Advanced related applications                    | 172        |
| Object versioning                                | 176        |
| Distributing multiple versions                   | 183        |
| <b>12 Cross-reference system</b>                 | <b>184</b> |
| Cross-reference file layout                      | 185        |
| Cross-reference commands                         | 188        |
| Working with object references                   | 190        |



|                                                                   |            |
|-------------------------------------------------------------------|------------|
| Working with user-defined references                              | 192        |
| <b>13 Promoting objects into independent ASPs</b>                 | <b>195</b> |
| <b>14 Managing IFS objects</b>                                    | <b>197</b> |
| Summary of TURNOVER® Lifecycle Manager changes for IFS            | 197        |
| Setting up TURNOVER® Lifecycle Manager to manage IFS objects      | 199        |
| Application defaults and rules                                    | 202        |
| Managing IFS objects                                              | 207        |
| Summary of commands for managing IFS objects                      | 207        |
| Checking out IFS objects                                          | 208        |
| Checking in IFS objects                                           | 209        |
| Promoting IFS objects                                             | 210        |
| Distributing IFS objects                                          | 214        |
| Finding TURNOVER® Lifecycle Manager information about IFS objects | 215        |
| IFS reference                                                     | 219        |
| Troubleshooting IFS support                                       | 222        |
| Optimizing Java programs                                          | 223        |
| Promoting files in the Windows file system                        | 226        |
| Promoting files in a UNIX file system                             | 227        |
| <b>15 Mirroring systems</b>                                       | <b>233</b> |
| <b>16 Setting up the mail daemon</b>                              | <b>236</b> |
| Considerations for using the mail daemon                          | 237        |
| Installing and configuring the mail daemon                        | 238        |
| Starting and ending the mail daemon manually                      | 242        |
| Determining where the autostart job will run                      | 243        |
| Multiple TURNOVER® Lifecycle Manager environments                 | 244        |
| Log settings                                                      | 246        |
| HTML message support                                              | 246        |
| <b>17 Mail integration specification</b>                          | <b>248</b> |
| Purpose                                                           | 248        |
| Open Authentication Specification v2.0                            | 261        |
| Microsoft Entra ID Service                                        | 270        |
| Initial Google Mail account authentication                        | 299        |
| Message handling                                                  | 304        |
| Running the TINBOXD mail monitor                                  | 309        |
| Appendix A – Flow diagram of mail process                         | 317        |
| <b>18 Managing physical and logical files</b>                     | <b>319</b> |
| New physical files                                                | 319        |
| Replacing physical files                                          | 321        |
| Deleting physical files                                           | 323        |
| New logical files                                                 | 323        |
| Replacing logical files                                           | 326        |
| Deleting logical files                                            | 327        |

|                                                               |            |
|---------------------------------------------------------------|------------|
| Duplicating files into Explode libraries                      | 327        |
| Why TURNOVER® Lifecycle Manager always compiles logical files | 328        |
| Distributing physical and logical files                       | 330        |
| Exception conditions                                          | 331        |
| <b>19 Using the CHGPF command</b>                             | <b>332</b> |
| Setting up CHGPF                                              | 333        |
| <b>20 Handling message IDs</b>                                | <b>337</b> |
| Defining type codes for message IDs                           | 337        |
| Assigning a message ID type code to an application            | 338        |
| Distributing message IDs                                      | 338        |
| Managing type code definitions                                | 339        |
| <b>21 Managing ILE programs</b>                               | <b>345</b> |
| Creating modules                                              | 345        |
| Creating programs                                             | 346        |
| Creating service programs                                     | 348        |
| Other ILE object types                                        | 349        |
| Work flow for managing ILE objects                            | 352        |
| Converting existing objects to ILE objects                    | 355        |
| <b>22 Menus, DFUs, and queries</b>                            | <b>360</b> |
| IBM i menus                                                   | 360        |
| IBM i DFUs                                                    | 366        |
| IBM i queries                                                 | 367        |
| <b>23 Managing SQL objects</b>                                | <b>368</b> |
| Managing tables                                               | 371        |
| Managing indexes and views                                    | 372        |
| Managing triggers                                             | 372        |
| Managing constraints                                          | 373        |
| Managing database security                                    | 374        |
| Managing functions and SQL stored procedures                  | 374        |
| Managing externally stored procedures and functions           | 378        |
| Managing materialized query tables                            | 383        |
| Managing programs that require SQL packages                   | 384        |
| Executing ad hoc SQL statements                               | 384        |
| Distributing SQL changes                                      | 386        |
| Object creation sequence                                      | 387        |
| Example: DDL source for a simple database                     | 388        |
| <b>24 Managing disk space</b>                                 | <b>390</b> |
| Delete the TURNOVER® Lifecycle Manager installation library   | 390        |
| <b>A Job descriptions</b>                                     | <b>393</b> |
| Job descriptions                                              | 393        |

|                                              |            |
|----------------------------------------------|------------|
| <b>B Configuration checklists</b>            | <b>405</b> |
| <b>C Commands</b>                            | <b>409</b> |
| Main menu options                            | 409        |
| Submenus                                     | 410        |
| TURNOVER® Lifecycle Manager commands         | 413        |
| <b>D Files and data areas</b>                | <b>431</b> |
| About TURNOVER® Lifecycle Manager files      | 431        |
| About TURNOVER® Lifecycle Manager data areas | 440        |

# 1 **TURNOVER<sup>®</sup> Lifecycle Manager Administrator's Guide**

This manual describes how the TURNOVER<sup>®</sup> Lifecycle Manager Administrator would plan, configure and administer TURNOVER<sup>®</sup> Lifecycle Manager on an IBM i computer. It assumes that you are familiar with:

- the IBM i operating system
- security administration
- the organization of your company's applications and databases
- Command Language (CL).

---

## **Choosing the TURNOVER<sup>®</sup> Lifecycle Manager administrator**

The TURNOVER<sup>®</sup> Lifecycle Manager Administrator should be well acquainted with work flow in your department; with methods and standards for controlling the flow of objects to production libraries; and, for a network of systems, issues related to the distribution and organization of objects on remote production systems.

The real objective of a TURNOVER<sup>®</sup> Lifecycle Manager system is to ensure that your department operates efficiently and that your production environment is stable and secure. TURNOVER<sup>®</sup> Lifecycle Manager does not ensure that you have adequately tested your programs. If used properly, it improves productivity and ensure that users cannot change objects casually and, when changes are made, a permanent audit trail exists. However, TURNOVER<sup>®</sup> Lifecycle Manager is not a substitute for an effective administrator.

The TURNOVER<sup>®</sup> Lifecycle Manager Administrator might perform any of the following tasks:

- Enter Change request and Problem Reports
- Evaluate Change request and Problem Reports
- Set up TURNOVER<sup>®</sup> Lifecycle Manager on your system
- Enter information about applications
- Maintain library and object authority

- Maintain TURNOVER® Lifecycle Manager user authority
- Maintain Programmer Information
- Verify plans for changes
- Review and certify test results
- Enter TURNOVER® Lifecycle Manager forms (a promotion request)
- Review TURNOVER® Lifecycle Manager forms entered by developers
- Run promotions and check the results
- Distribute promotions to remote systems
- Authorize emergency program fixes
- Produce reports and monitor change history
- Audit change history on your system.

## 2 Application planning

Many of the procedures described in this section of the document are based on the TURNOVER® Lifecycle Manager Administrator module. Install and configure that software; enroll yourself as a TURNOVER® Lifecycle Manager user; and give yourself authority for the TURNOVER® Lifecycle Manager application models 01M – 09M.

You will need to have a general knowledge of the IBM i operating system; are familiar with security administration; and understand the organization of your company's applications and databases; and have experience of Command Language (CL).

### See

*Overview of TURNOVER® Lifecycle Manager* below.

*'Application definitions' on page 32*

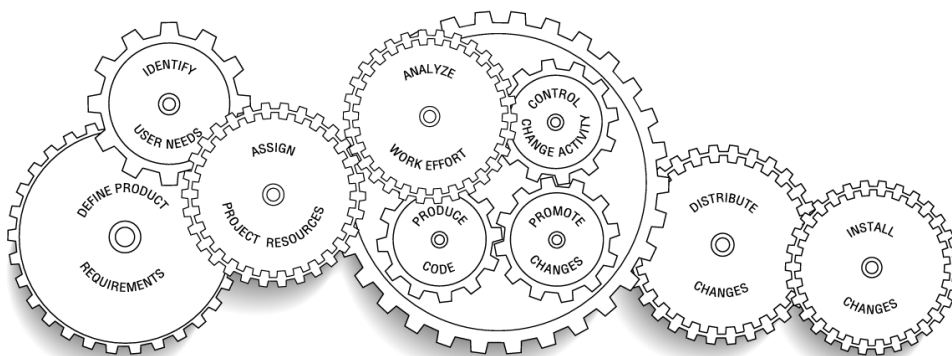
*'For more information, see the TURNOVER® Lifecycle Manager Developer's Guide.' on page 37*

*'Configuration checklists' on page 405*

---

### Overview of TURNOVER® Lifecycle Manager

With capabilities such as Helpdesk, TURNOVER® Lifecycle Manager Projects, Audit reporting, Synchronizer, and Programmer's Worklist Manager (PWM) combined with a set of interface products, the TURNOVER® Lifecycle Manager system redefines change management into a framework for software development.



TURNOVER® Lifecycle Manager has many component parts, all integrated into one system. These include:

- Helpdesk and online user entry of change requests
- Project management system
- Source control management
- Analysis tools such as a built-in cross-referencing facility and version management facility
- Programmer's Worklist Manager facility: a PDM-like facility combining PDM and change management functions on one panel
- Promotion management
- Object and source archives
- Distribution and remote installation management
- Project time and cost reporting
- Audit and history reporting
- Synchronizer vendor software synchronization with advanced source compare/merge utilities
- CASE & other development tool interfaces.

You decide which features to implement based on your needs.

## Change management

Change management defines the role of most information systems software development organizations. To meet the changing business requirements of your enterprise, the information systems organization must change the software that implements those requirements. As technology changes, the tools used to implement those requirements also change. In the midst of all this change, you need a stable framework from which to manage and fulfill your responsibilities. TURNOVER® Lifecycle Manager is that framework.

Think of TURNOVER® Lifecycle Manager as *your* departmental application. Whether you are a systems manager, administrator or developer, your role can best be described as *change agent*. Invariably, implementing something new means defining an evolutionary path from what is, to what needs to be. How could you hope to do that well without a comprehensive system?

This section introduces change management concepts and the ways that TURNOVER® Lifecycle Manager helps you become a more effective *Change Agent*. It assumes that you are the person who will set up your software project tracking and development environments, and that you will also work as a developer.

The person who sets up projects and applications and enrolls developers in TURNOVER® Lifecycle Manager's security files is the *TURNOVER® Lifecycle Manager administrator*.

---

## How TURNOVER® Lifecycle Manager improves productivity

These examples illustrate one or more features of TURNOVER® Lifecycle Manager and show how you might approach a particular problem or task.

### **Example: Add a field to a physical file**

Adding a field to a physical file requires recompiling all of the logical files and programs that refer to the physical file.

**Without** TURNOVER® Lifecycle Manager, you would:

- 1 Research all of the program and logical file source for references to the PF
  - 2 Write down a list of these objects
  - 3 Copy the source from production to a development library
  - 4 Change the source
  - 5 Recompile the PF in development
  - 6 Move the old PF to another library
  - 7 Copy the old PF source to another source file (archive)
  - 8 Copy the new PF source to the production library
  - 9 Delete all of the related LFs
  - 10 Create the new PF
  - 11 Recreate all the related LFs
  - 12 Recompile all of the programs that refer to the PF or LFs
  - 13 Copy the data from the old file to the new file
  - 14 Set object ownership and grant object authority appropriately.
- ...all without making a mistake.

**With** TURNOVER® Lifecycle Manager, you would:

- 15 Check out the PF source.
- 16 Change the source.
- 17 Compile to new PF in development.
- 18 Build a promotion request form and add the related LFs and programs automatically.
- 19 Submit the form.



TURNOVER® Lifecycle Manager takes care of the rest – without making an error.

If you want to undo your changes after they are installed:

- Without TURNOVER® Lifecycle Manager, you would have to reverse all of the former steps, by hand, one at a time.
- With TURNOVER® Lifecycle Manager, you simply build and run a recovery form. The rest is automatic.

### **Example: Make an emergency change**

Suppose its 3 a.m. on Sunday morning, and you need to change a program that John is working on.

Without TURNOVER® Lifecycle Manager, you would have to:

- 1 Sign on as QSECOFR or to a profile with \*ALLOBJ authority
- 2 Copy the production source to another source archive file
- 3 Make the change and test it
- 4 Copy the source back to the production source file
- 5 Compile the object in production
- 6 Set object ownership and grant object authority appropriately
- 7 Remember to tell John you had made the emergency change...
- 8 Hope that John makes the same change in test

With TURNOVER® Lifecycle Manager, you would:

- 9 Check out the object using an emergency change application
- 10 Make the necessary change to the source in a development library
- 11 Run the emergency change form to promote the change and notify the programmer

TURNOVER® Lifecycle Manager automatically takes care of the rest of the process, including notifying John to remind him to make the same change.

### **More examples**

Sometimes, even simple situations can consume time, especially if they occur frequently.

- You need to check for an object lock on the entry operator file before installing changes, every time one of several objects changed. With TURNOVER® Lifecycle Manager, you can be sure you do not disturb active users.
- You need to change a PF with 8 members, all with live data. With TURNOVER® Lifecycle Manager, your physical files members will be preserved whenever you make a change to a file.

- You want to change the same database file format twenty times in twenty different company libraries. With TURNOVER® Lifecycle Manager, your change can ripple through a list of database libraries, effortlessly.
- You need to identify what objects were changed when a problem was fixed at a certain date in the past. With TURNOVER® Lifecycle Manager, you can easily retrieve a list of objects related to any historical task.
- You want to *verify all of the source and objects* before promoting them. With TURNOVER® Lifecycle Manager, you can run an option to pre-check your changes, before submitting the TURNOVER® Lifecycle Manager job.
- Suppose you want to promote 200 programs and related objects at 3 a.m. With TURNOVER® Lifecycle Manager's many verification checks and automatic rollback, your likelihood of success is greater, *and* if an error does occur, rollback is automatic.

As a developer, you probably want to spend your time developing instead of copying source, compiling objects, setting object authority, and fixing installation errors. With TURNOVER® Lifecycle Manager, your role ends when you build the promotion request form; a change administrator can take over from there.

---

## Setting up TURNOVER® Lifecycle Manager

Optimally, TURNOVER® Lifecycle Manager and its object history database should provide the *only* means through which source and objects can be added, replaced or deleted from a production library.

To ensure that this is the case, you must establish security on your production objects. If you do not, TURNOVER® Lifecycle Manager will still ease the process of managing your production libraries. Because you might still have a security exposure, you risk losing many of TURNOVER® Lifecycle Manager benefits; however, TURNOVER® Lifecycle Manager's audit reports and messages inform you of possible security breaches.

For more information about TURNOVER® Lifecycle Manager and IBM i security, see [‘Security recommendations’ on page 149](#).

---

## Security

TURNOVER® Lifecycle Manager assumes that your production libraries and objects are already secure. It complements and enhances IBM i security, but cannot operate without it.

TURNOVER® Lifecycle Manager also assumes that passwords for user profiles that are the owners of your production objects (such as QSECOFR) are either not known by users other than the system manager, or that their passwords are set to \*NONE. If programmers or operators can sign on as QSECOFR, or if they run procedures that operate under the authority of QSECOFR (a

favorite trick of programmers to circumvent security), then they can still modify objects in production and bypass TURNOVER® Lifecycle Manager procedures. If someone has \*ALLOBJ authority, no application or system can prevent them from getting at production files and programs.

You must promote and enforce good security practices. If you do not, production libraries can become corrupted and, other than running TURNOVER® Lifecycle Manager Audit report periodically, you have no reliable means of auditing to determine how a change was made.

This section describes some approaches you can take when setting up TURNOVER® Lifecycle Manager to model your operations, from the most basic test-to-production configuration to complex multiple test and production library configurations on multiple systems.

For information about more advanced object management techniques associated with maintaining multiple versions of a software application, see *'Managing multiple versions' on page 161*.

---

## About library structure

When you begin modeling your software change environments using TURNOVER® Lifecycle Manager application definitions, set up your application definitions with separate libraries for source files, program objects and database files. This approach has these benefits:

- You can save the application libraries independently, with different save frequencies for data than for programs.
- You can remove source from the system if you need space.
- You can more easily secure data files differently than programs.
- You can substitute different data libraries in the library list while continuing to use the same set of programs.
- You can more easily install applications on other computers.
- You can assign programming responsibility by library.

---

## Establishing source archive policies

When you are deciding whether or not, and how to, archive the source code for your software applications, consider these points:

- Keep it simple. The less manual intervention that is required, the better.
- Do you have any reason to archive source at more than one application level?
- Perhaps you have a need to maintain both QA and Production source archives.
- How long should you retain archived source? Consider factors such as company policy, development requirements, and system resources.

For example, if you are keeping source archives at both the QA and production levels of your application, purge your QA archive more aggressively than your production archive.

- How will you structure your files?

You can either change your archive file name on a periodic basis and save the old file off line after some appropriate time span, or use TURNOVER® Lifecycle Manager's `TRGZARCF` command (Reorganize Archived Source File) to move archived members into other files that you then save off line, leaving your archive file name unchanged.

- If you want to archive according to a number of versions saved, use the `TRGZARCF` command to move members older than a specific version (or date) to another file, which you can retain on line for some period and then save off after the retention period lapses.

Remember: these concepts describe *only source* archiving; TURNOVER® Lifecycle Manager archives objects differently.

## Archiving examples

### Periodic global archiving

Set global default archive file to `ARCHyyyyymm`. Each month, change the global archive file name to reflect the month, and retain the old monthly archive files for some appropriate time span (24 months, for example). When the time span expires, save the files off line using the Source Archive Files utility in Administration Utilities.

If you do not need to change or track archive files on a monthly basis, you can apply this concept to quarterly or annual time frames.

This is a very simple scenario that has minimal maintenance overhead associated with it.

### Periodic single-level application archiving

Set the archive source file name at an appropriate level of each application definition to something that reflects the application name and archiving period. Each period, change the archive name as necessary for each application. Retain the old application archives as needed and save them off line using Administration Utilities when the retention period expires.

This scenario organizes your source archives neatly by application, which can aid your later source retrieval. However, if you have a lot of applications, the overhead of changing each application's archive file name on a periodic basis might be unacceptable. This is a case where the file reorganize capability might help you avoid redundant application maintenance.

### Periodic combination archiving

Maybe you do not mind storing the archived source for some applications together, but you have some you definitely want to keep separate. Just set up the global default for the ones you can lump, and override the global default for the ones you want separate at the application level.

## Non-periodic global or application archiving

One reason for changing TURNOVER® Lifecycle Manager's archiving process was to address file member limitations. The ability to split your archives off by application lets you distribute the member load across files, extending the life of your archives and saving you maintenance time. If this is not an issue for you, you can archive all source to either a global or application location, and attend to retention and storage at your leisure, whether by changing archive file names or by reorganizing existing files.

## More information about archiving

For more information about archiving, see:

- Chapter 5 of the *TURNOVER® Lifecycle Manager User Guide*
- 'Work with Source Archive Files' in the *TURNOVER® Lifecycle Manager User Guide*.
- TURNOVER® Client - Administrator online help.

---

## Where do TURNOVER® Lifecycle Manager projects come in?

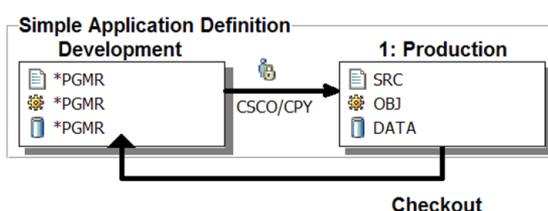
If you use the TURNOVER® Lifecycle Manager Project system, every change you make will likely be associated with a TURNOVER® Lifecycle Manager project task. You identify the project task when you determine what object you need to change. This association is maintained for you throughout the process: when you check out objects, as you work on them, when you promote them, and (at some future time) when you are reporting on changes you made. Also, your task status can be automatically updated as you perform checkout duties and as you promote changes to test, quality assurance, distribute changes, and so on.

---

## Designing your TURNOVER® Lifecycle Manager change environments

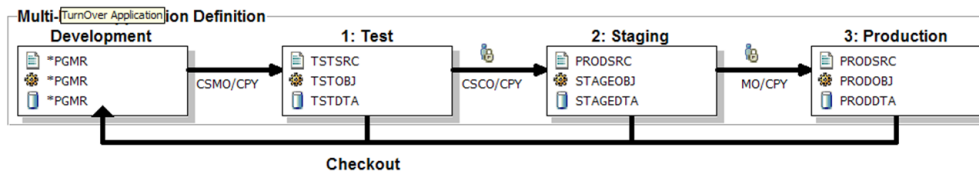
### Example 1: A simple change environment

For a software application that is not mission-critical and changes only occasionally, one application level might be enough.



## Example 2: A more complex change environment

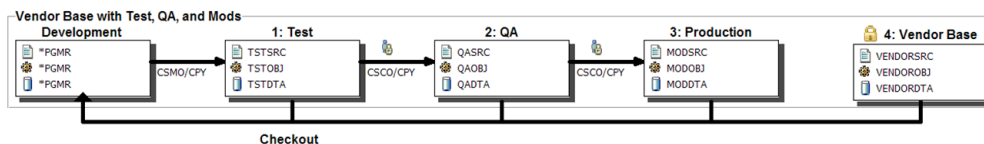
For a more sensitive software application, which is mission critical and for which change are distributed to hundreds of users, three or four application definition levels might be more appropriate.



In this example, changes promoted into level 2 are compiled into a staging library. With this application setup, you might run your level 2 forms during the day, then promote the Level 3 changes into production at night using the speedier MO (Move Object) method. This is a popular method with organizations that provide services around the clock, such as hospitals.

## Example 3: An environment for vendor software

Suppose another of your software applications is supplied by a software vendor, and your company makes custom modifications to the vendor base. In this scenario, you need to ensure that your changes remain separate from the vendor's base. In your TURNOVER® Lifecycle Manager application definition, you define the application level containing the vendor's base code as *Permanently Locked*.

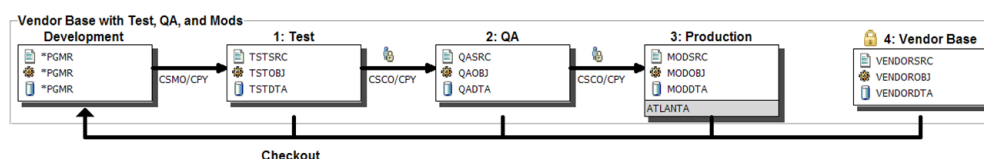


You can have any number of permanently locked levels, as is the case where the vendor sends periodic PTF changes.

## Example 4: Distributing changes to your vendor software

To distribute changes to one or more computers or partitions, you can use a remote version of TURNOVER® Lifecycle Manager running on that system. Generally, you distribute changes from the production level after objects are promoted there; however, you can distribute from any level if you choose.

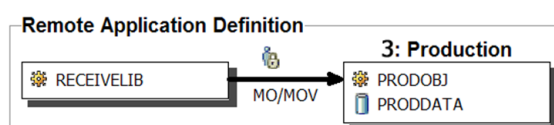
(For more information about distributing changes, see 'Distributing to Logical Partitions' in the *TURNOVER® Lifecycle Manager User Guide*.)



Once changes are successfully promoted to level 3, TURNOVER® Lifecycle Manager starts a distribution job that bundles the changes into a save file and sends them to the remote computer named ATLANTA. There, TURNOVER® Lifecycle Manager receives the save file and install the changes, either immediately or later as a timed job.

**Important** A remote application definition for the sending application level must exist on the production computer.

For the application illustrated above, you need a remote application definition like this on the ATLANTA system:



Most companies distribute objects only. However, you can distribute source with objects and compile on the production computer. The names of the target libraries do not have to match the library names on your development computer. In fact, all defaults are assigned by the receiving application definition.

You can also distribute at multiple application levels. For example, you might want to distribute changes that you have promoted into a quality assurance location to a similar environment on your production system, and have your users test and certify the changes there before promoting them into production (not shown). This way, you can maintain a *training/testing environment* on your production computer.

If you need to manage several releases and/or maintain customized version of software for several locations or customers, see [‘Managing multiple versions’ on page 161](#).

## Modeling with application definition models

TURNOVER® Lifecycle Manager includes some pre-configured application definition models that you can copy and customize. You can also create new application definitions by using the TURNOVER® Client.

### To define your application

- 1 Review the schematics, and then choose one that fits best.
- 2 Diagram your own setup on paper.
- 3 If you intend to use the TURNOVER® Lifecycle Manager project system, use the TURNOVER® Client to create a project and add tasks.
- 4 Test the application definitions thoroughly.

## Before you begin

- 1 If you plan to distribute changes, install TURNOVER® Lifecycle Manager Remote on the other computers in your network, and then *define those systems on your development computer*. Later, when you have created your remote application definitions, you can use the Remote Configuration Wizard to send your system and remote application definitions to your remote computers.

To define systems in the TURNOVER® Client, expand the Administration section, right-click **Distribution Systems**, and then click **New System**.

- 2 Copy and customize the model application definitions to reflect your setup requirements.

In the TURNOVER® Client, expand the Applications section, right-click an application model, and then click **Copy application**.

- 3 If your software applications require a programming language other than RPG or object types that are not included in the models, update the type code list in your customized model as necessary.

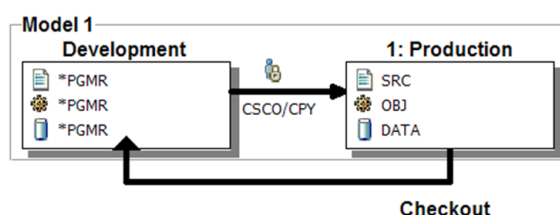
In edit mode in the Application Designer, right-click an application level, and then click **Add type codes**.

- 4 Enroll TURNOVER® Lifecycle Manager users.

In the TURNOVER® Client, expand the Administration section, right-click **Users**, and then click **New User**.

TURNOVER® Lifecycle Manager's model application definitions are set for a typical RPG development environment and assume a single computer. Performing these steps will save time later.

## Model 1 (01M)



In this model, source is copied from development to production, and production objects are compiled using the copied source.

- 1 Source for programs and other objects is checked out to a Programmer's development library (or a common development or project library).

New object names are reserved in TURNOVER® Lifecycle Manager.

A naming convention such as when object names must start with "PROD" or "XXX", may be enforced by application and object type.



- 2 Programmer makes changes to source, and compiles and tests it in his/her library.
- 3 Programmer builds TURNOVER® Lifecycle Manager form and runs error-check.
- 4 Form is approved.
- 5 Form is run. Source is copied into SRC library and compiled into OBJ library.

Library list for level 1 form is:

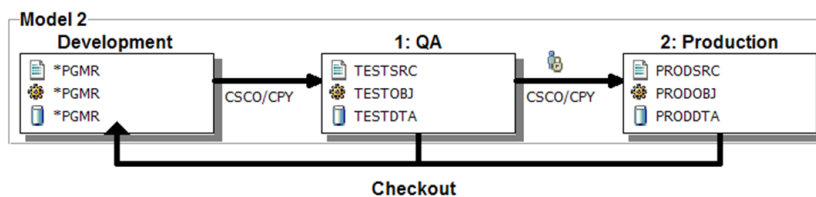
SRC  
OBJ  
DATA

IFS objects are moved or copied.

Programmer's development library should *not* be included in the library list.

The programmer's library contains only *changed* objects. They are deleted after the TURNOVER® Lifecycle Manager form runs.

## Model 2 (02M)



In this model, source is copied to QA from development, and objects are compiled in QA from the copied source. Production objects are compiled using source copied from QA.

- 1 Source for programs and other objects is checked out to a Programmer's development library (or a common development or project library).

New object names are reserved in TURNOVER® Lifecycle Manager.

A naming convention such as when object names must start with "PROD" or "XXX", may be enforced by application and object type.

- 2 Programmer makes changes to source, and compiles and tests it in his/her library.
- 3 Programmer builds TURNOVER® Lifecycle Manager level 1 form and runs error-check.
- 4 Level 1 form is approved.
- 5 Level 1 form is run. Source is copied into TESTSRC library and compiled into TESTOBJ. IFS objects are moved or copied.

Library list for level 1 form is:

TESTSRC  
TESTOBJ  
TESTDTA  
PRODSRC  
PRODOBJ  
PRODDTA

Programmer's development library should not be included in the library list.

TESTOBJ and TESTDTA contain only changed objects.

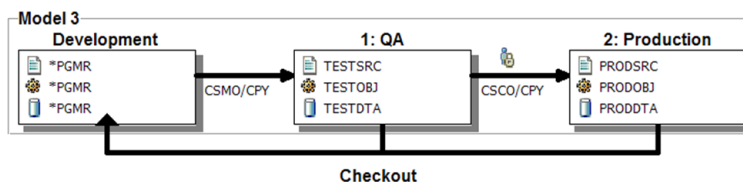
- 6 TURNOVER® Lifecycle Manager administrator copies one or more level 1 forms to level 2 (preparing them to run for level 2).
- 7 Level 2 form is approved and run. TURNOVER® Lifecycle Manager copies source into PRODSRC and compiles objects into PRODOBJ and PRODDTA. LFs are recreated over PFs. Old source is archived. Old object is archived (optional). Source and object are recorded as having been checked in. IFS objects are moved or copied.

Library list for level 2 run is:

PRODSRC  
PRODOBJ  
PRODDTA  
:  
:

This library list is all that should be needed to create objects unless some programs refer to objects in other libraries.

## Model 3 (03M)



In this model, the source is copied from development into QA, and QA objects are moved in from development. Production objects are compiled using source copied from QA.

- 1 Source for programs and other objects is checked out to a Programmer's development library (or a common development or project library). New object names are reserved in TURNOVER® Lifecycle Manager.

A naming convention such as when object names must start with "PROD" or "XXX", may be enforced by application and object type.

- 2 Programmer makes changes to source and compiles and tests it in his/her library.
- 3 Programmer builds TURNOVER® Lifecycle Manager level 1 form and runs error-check.

- 4 Level 1 form is approved.
- 5 Level 1 form is run. Source is copied into TESTSRC library and object is moved into TESTOBJ. IFS objects are moved or copied.

Library list for level 1 form is:

```
TESTSRC
TESTOBJ
TESTDTA
PRODOBJ
:
```

Programmer's development library should *not* be included in the library list.

TESTOBJ and TESTDTA contain only *changed* objects.

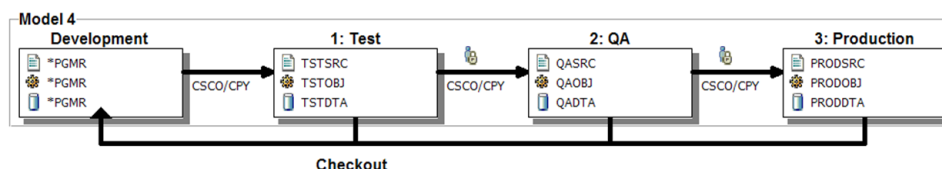
- 6 TURNOVER® Lifecycle Manager administrator copies one or more level 1 forms to level 2 (preparing them to run for level 2).
- 7 Level 2 form is approved and run. TURNOVER® Lifecycle Manager copies source into PRODSRC and compiles objects into PRODOBJ and PRODDTA. IFS objects are moved or copied. LFs are recreated over PFs. Existing data is copied to new files. Old source is archived. Old object is archived (optional). Source is recorded as having been checked in.

Library list for level 2 run is:

```
PRODSRC
PRODOBJ
PRODDTA
:
```

This library list is all that should be needed to create objects unless some programs refer to objects in other libraries.

## Model 4 (04M)



In this model, Test objects are compiled from source that is copied in from Development. QA and Production objects are both created from source that is copied in from the previous level.

- 1 Source for programs and other objects is checked out to a Programmer's development library (or a common development or project library). New object names are reserved in TURNOVER® Lifecycle Manager.

A naming convention such as when object names must start with "PROD" or "XXX", may be enforced by application and object type.

- 2 Programmer makes changes to source and compiles and tests it in his/her library.
- 3 Programmer builds TURNOVER® Lifecycle Manager level 1 form and runs error-check.
- 4 Level 1 form is approved and run. TURNOVER® Lifecycle Manager copies source into TSTSRC library and compiles objects into TSTOBJ and TSTDTA. IFS objects are moved or copied.

Library list for level 1 form is:

TSTSRC QADTA

TSTOBJ PRODSRC

TSTDTA PRODOBJ

PRODDTA

QASRC:

QAOBJ:

Programmer's development library should *not* be included in the library list.

TSTOBJ and TSTDTA contain only *changed* objects.

- 5 TURNOVER® Lifecycle Manager administrator copies one or more level 1 forms to level 2 (preparing them to run for level 2).
- 6 Level 2 form is approved.
- 7 Level 2 form is run. TURNOVER® Lifecycle Manager copies source into QASRC and compiles objects into QAOBJ and QADTA. IFS objects are moved or copied.

Library list for level 2 run is:

QASRC

QAOBJ

QADTA

PRODSRC

PRODOBJ

PRODDTA

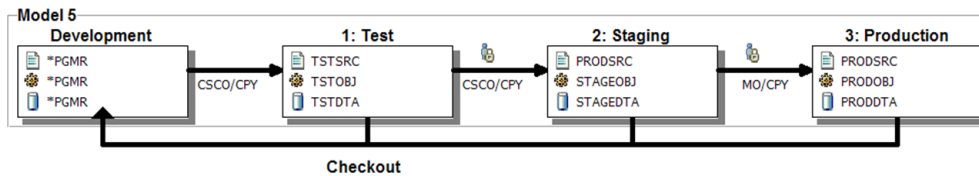
- 8 TURNOVER® Lifecycle Manager Administrator copies one or more Level 2 forms to Level 3 (preparing them to run for Level 3).
- 9 Level 3 form is approved and run. TURNOVER® Lifecycle Manager copies source into PRODSRC and compiles objects into PRODOBJ and PRODDTA. IFS objects are moved or copied. LFs are recreated over PFs. Old source is archived. Old object is archived (optional). Source is recorded as having been checked in.

Library list for Level 3 run is:

PRODSRC  
PRODOBJ  
PRODDTA  
:

This library list is all that should be needed to create objects unless some programs refer to objects in other libraries.

## Model 5 (05M)



This model uses a pre-production staging area. Objects in Test are compiled from source copied in from Development. Staging objects area created in Staging, from Production source. Production objects are moved or copied in from Staging.

- 1 Source for programs and other objects is checked out to a Programmer's development library (or a common development or project library). New object names are reserved in TURNOVER® Lifecycle Manager.

A naming convention such as when object names must start with "PROD" or "XXX", may be enforced by application and object type.

- 2 Programmer makes changes to source and compiles and tests it in his/her library.
- 3 Programmer builds TURNOVER® Lifecycle Manager level 1 form and runs error-check.
- 4 Level 1 form is approved and run. TURNOVER® Lifecycle Manager copies source into TSTSRC library and compiles objects into TSTOBJ and TSTDTA. IFS objects are moved or copied.

Library list for level 1 form is:

TSTSRCPRODSRC  
TSTOBJPRODOBJ  
TSTDTAPRODDTA  
:  
STAGEOBJ:  
STAGEDTA

Programmer's development library should *not* be included in the library list.

TSTOBJ and TSTDTA contain only *changed* objects.

- 5 TURNOVER® Lifecycle Manager administrator copies one or more level 1 forms to level 2 (preparing them to run for level 2).
- 6 Level 2 form is approved.

- 7 Level 2 form is run. Copy source into PRODSRC and compile objects into STAGEOBJ and STAGEDTA. IFS objects are moved or copied.

Library list for level 2 run is:

```
PRODSRC
STAGEOBJ
STAGEDTA
PRODOBJ
PRODDTA
```

- 8 TURNOVER® Lifecycle Manager Administrator copies one or more Level 2 forms to Level 3 (preparing them to run for Level 3).
- 9 Level 3 form is approved and run. Objects are moved into PRODOBJ and PRODDTA. IFS objects are moved or copied. LFs are recreated over PFs. Old source is archived. Old object is archived (optional). Source is recorded as having been checked in. Data is copied to new files.

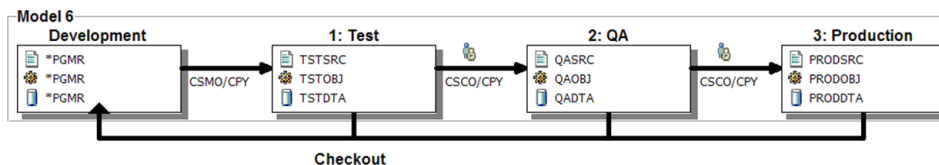
Library list for Level 3 run is:

```
PRODSRC
PRODOBJ
PRODDTA
:
```

This library list is all that should be needed to create objects unless some programs refer to objects in other libraries.

The source library at level 2 and 3 are the same name; This is intentional.

## Model 6 (06M)



In this model, the objects at all levels are compiled from source that is copied in from the previous level.

- 1 Source for programs and other objects is checked out to a Programmer's development library (or a common development or project library). New object names are reserved in TURNOVER® Lifecycle Manager.

A naming convention such as when object names must start with "PROD" or "XXX", may be enforced by application and object type.

- 2 Programmer makes changes to source and compiles and tests it in his/her library.
- 3 Programmer builds TURNOVER® Lifecycle Manager level 1 form and runs error-check.

- 4 Level 1 form is approved and run. TURNOVER® Lifecycle Manager copies source into TSTSRC library and moves objects into TSTOBJ and TSTDТА. IFS objects are moved or copied.

Library list for level 1 form is:

TSTSRCPRODSRC

TSTOBJPRODOBJ

TSTDТАPRODDТА

QASRC

QAOBJ:

QADТА:

Programmer's development library should *not* be included in the library list.

TSTOBJ and TSTDТА contain only *changed* objects.

- 5 TURNOVER® Lifecycle Manager administrator copies one or more level 1 forms to level 2 (preparing them to run for level 2).
- 6 Level 2 form is approved and run. TURNOVER® Lifecycle Manager copies source into QASRC and compiles objects into QAOBJ and QADТА. IFS objects are moved or copied.

Library list for level 2 run is:

QASRC

QAOBJ

QADТА

PRODSRC

PRODOBJ

PRODDТА

:

- 7 TURNOVER® Lifecycle Manager Administrator copies one or more Level 2 forms to Level 3 (preparing them to run for Level 3).
- 8 Level 3 form is approved and run. TURNOVER® Lifecycle Manager copies source into PRODSRC and compiles objects into PRODOBJ and PRODDТА. IFS objects are moved or copied. LFs are recreated over PFs. Old source is archived. Old object is archived (optional). Source is recorded as having been checked in.

Library list for Level 3 run is:

PRODSRC

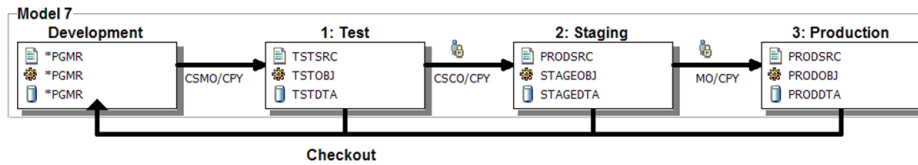
PRODOBJ

PRODDТА

:

This library list is all that should be needed to create objects unless some programs refer to objects in other libraries.

## Model 7 (07M)



In this model, Test source is copied in from Development, while the objects are moved. At the Staging level, source is copied into Production and Staging objects are created from the Production source. Production objects are moved in from Staging.

- 1 Source for programs and other objects is checked out to a Programmer's development library (or a common development or project library). New object names are reserved in TURNOVER® Lifecycle Manager.

A naming convention such as when object names must start with "PROD" or "XXX", may be enforced by application and object type.

- 2 Programmer makes changes to source and compiles and tests it in his/her library.
- 3 Programmer builds TURNOVER® Lifecycle Manager level 1 form and runs error-check.
- 4 Level 1 form is approved and run. TURNOVER® Lifecycle Manager copies source into TESTSRC library and moves objects into TESTOBJ. IFS objects are moved or copied.

Library list for level 1 form is:

```
TESTSRC
TESTOBJ
TESTDTA
:
```

Programmer's development library should *not* be included in the library list.

TESTOBJ and TESTDTA contain only *changed* objects.

- 5 TURNOVER® Lifecycle Manager administrator copies one or more level 1 forms to level 2 (preparing them to run for level 2).
- 6 Level 2 form is approved and run. TURNOVER® Lifecycle Manager copies source into PRODSRC and compiles objects into STAGEOBJ and STAGEDTA. IFS objects are moved or copied. Old source is archived.

Library list for level 2 run is:

```
PRODSRC
STAGEOBJ
STAGEDTA
PRODSRC
PRODOBJ
PRODDTA
:
```



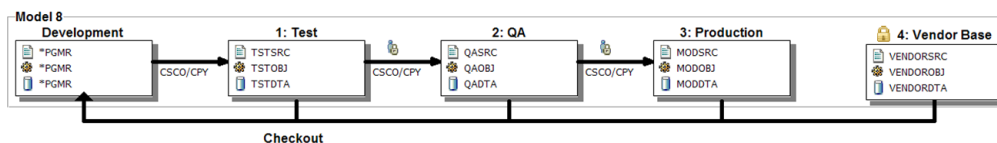
- 7 TURNOVER® Lifecycle Manager Administrator copies one or more Level 2 forms to Level 3 (preparing them to run for Level 3).
- 8 Level 3 form is approved and run. TURNOVER® Lifecycle Manager moves objects into PRODOBJ and PRODDTA. IFS objects are moved or copied. LFs are recreated over PFs. Old object is archived (optional). Source is recorded as having been checked in. Existing data is copied to new files.

Library list for Level 3 run is:

PRODSRC  
PRODOBJ  
PRODDTA  
:

This library list is all that should be needed to create objects unless some programs refer to objects in other libraries.

## Model 8 (08M): Vendor package model A



In this model, source and objects are checked out of the Vendor base. Source is copied and objects are compiled at all levels. Changed source and objects remain at the Production level and Vendor source is preserved in the permanently locked level.

- 1 Source for programs and other objects is checked out to a Programmer's development library (or a common development or project library). TURNOVER® Lifecycle Manager looks first in MODSRC and then in VENDORSRC and copies it to development library. New object names are reserved in TURNOVER® Lifecycle Manager.

A naming convention such as when object names must start with "PROD" or "XXX", may be enforced by application and object type.

- 2 The programmer makes changes to source and compiles and tests it in their library.
- 3 The programmer builds TURNOVER® Lifecycle Manager level 1 form and runs error-check.
- 4 Level 1 form is approved and run. TURNOVER® Lifecycle Manager copies source into TESTSRC library and compiles objects into TESTOBJ and TESTDTA. IFS objects are moved or copied.

Library list for level 1 form is:

TESTSRC  
MODSRC  
TESTOBJ  
MODOBJ  
TESTDTA

MODDTA  
QASRC  
ENDORSRC  
QAOBJ  
VENDOROBJ  
QADTA  
VENDORDTA

The programmer's development library should *not* be included in the library list. TESTOBJ and TESTDTA contain only objects being changed.

- 5 TURNOVER® Lifecycle Manager administrator copies one or more level 1 forms to level 2 (preparing them to run for level 2).
- 6 Level 2 form is approved and run. TURNOVER® Lifecycle Manager copies source into QASRC and compiles objects into QAOBJ and QADTA. IFS objects are moved or copied.

Library list for level 2 run is:

QASRC  
QAOBJ  
QADTA  
MODSRC  
MODOBJ  
MODDTA  
VENDORSRC  
VENDOROBJ  
VENDORDTA

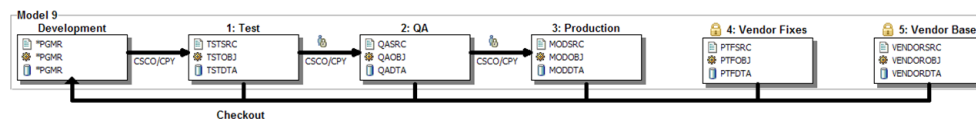
- 7 TURNOVER® Lifecycle Manager Administrator copies one or more Level 2 forms to Level 3 (preparing them to run for Level 3).
- 8 Level 3 form is approved and run. TURNOVER® Lifecycle Manager copies source into MODSRC and compiles objects into MODOBJ and MODDTA. IFS objects are moved or copied. LFs are recreated over PFs. Old source is archived. Old object is archived (optional). Source is recorded as having been checked in, even if it had been copied from VENDORSRC.

Library list for Level 3 run is:

MODSRC  
MODOBJ  
MODDTA  
VENDORSRC  
VENDOROBJ  
VENDORDTA

This library list is all that should be needed to create objects unless some programs refer to objects in other libraries.

## Model 9 (09M): Vendor package model B



In this revised version of the Vendor Base model, a second locked level is added where Vendor-supplied fixes are stored. Like the original Vendor model, source is copied and then compiled at all levels, but promotion stops at the Production level, and original Vendor source and Vendor fixes are preserved in the permanently locked levels.

- 1 Source for programs and other objects is checked out to a Programmer's development library (or a common development or project library). TURNOVER® Lifecycle Manager looks in MODSRC first, then in PTFSRC, then in VENDORSRC, and copies it to development library. New object names are reserved in TURNOVER® Lifecycle Manager.

A naming convention such as when object names must start with "PROD" or "XXX", may be enforced by application and object type.

- 2 Programmer makes changes to source and compiles and tests it in his/her library.
- 3 Programmer builds TURNOVER® Lifecycle Manager level 1 form and runs error-check.
- 4 Level 1 form is approved and run. TURNOVER® Lifecycle Manager copies source into TESTSRC library and compiles objects into TESTOBJ and TESTDTA. IFS objects are moved or copied.

Library list for level 1 form is:

TESTSRC  
MODOBJ  
TESTOBJ  
MODDTA  
TESTDTA  
PTFSRC  
QASRC  
PTFOBJ  
QAOBJ  
PTFDTA  
QADTA  
VENDORSRC  
VENDOROBJ  
VENDORDTA  
MODSRC

Programmer's development library should *not* be included in the library list.

TESTOBJ and TESTDTA contain only objects being changed.

- 5 TURNOVER® Lifecycle Manager administrator copies one or more level 1 forms to level 2 (preparing them to run for level 2).

- 6 Level 2 form is approved and run. TURNOVER® Lifecycle Manager copies source into QASRC and compiles objects into QAOBJ and QADTA. IFS objects are moved or copied.

Library list for level 2 run is:

QASRC  
PTFSRC  
QAOBJ  
PTFOBJ  
QADTA  
PTFDTA  
VENDORSRC  
VENDOROBJ  
VENDORDTA  
MODSRC  
MODOBJ  
MODDTA

- 7 TURNOVER® Lifecycle Manager Administrator copies one or more Level 2 forms to Level 3 (preparing them to run for Level 3).
- 8 Level 3 form is approved and run. TURNOVER® Lifecycle Manager copies source into MODSRC and compiles objects into MODOBJ and MODDTA. IFS objects are moved or copied. LFs are recreated over PFs. Old source is archived. Old object is archived (optional). Source is recorded as having been checked in, even if had been copied from PTFSRC or VENDORSRC.

Library list for Level 3 run is:

MODSRC  
PTFDTA  
MODOBJ  
VENDORSRC  
MODDTA  
VENDOROBJ  
PTFSRC  
VENDORDTA  
PTFOBJ  
:

This library list is all that should be needed to create objects unless some programs refer to objects in other libraries.

---

## Application definitions

Application definitions are the core of a TURNOVER® Lifecycle Manager workflow.

## What is a software application?

A software application is a collection of related objects that, taken together, perform a specific business function. An example might be financial software applications such as accounts payable, order entry, general ledger and accounts receivable.

You might choose to manage these collectively and define only one application definition in TURNOVER® Lifecycle Manager for all of them, or you might manage each one separately. Alternatively, you might create a “DBA” application definition to manage all data object types, and one or more other application definitions to manage all non-data, program-related object types.

## What is a TURNOVER® Lifecycle Manager application definition?

You use TURNOVER® Lifecycle Manager application definitions to model the testing and production environments for your software applications, and to define pre-set options that support the usual procedures to be followed when your programmers check out source and build TURNOVER® Lifecycle Manager forms.

The values associated with an application definition are called *application defaults*. Application defaults let you set the rules. For example, the TURNOVER® Lifecycle Manager application definition for any given software application that you manage can specify test (from) and production (target) libraries, enforce object naming standards (optional), define rules for promoting objects, and specify which object types are appropriate.

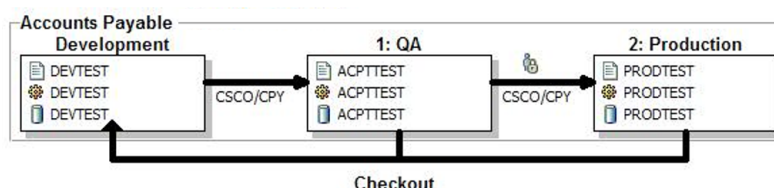
Application definitions control how TURNOVER® Lifecycle Manager processes your software changes. With TURNOVER® Lifecycle Manager, you can manage each of your software applications differently, according to its own particular requirements. For example, for some software applications you manage, you might use intermediate quality assurance test libraries, while for others you might not. Some of your software applications might be vendor-supplied, whereas others are developed in-house.

## About application diagrams

An application diagram helps you to visualize what a application definition is set up to do.

The diagram shows a example application schematic. TURNOVER® Lifecycle Manager can manage software applications in many different ways.

## Understanding an application diagram



Each box represents one set of libraries. The production libraries are on the right, development libraries on the left. You can have any number of intermediate libraries, depending on your company's requirements. (In this case, ACPTTEST stands for Acceptance Test.)

Checkout moves source from *right to left*; promotions move source and objects from *left to right*. The development libraries can be a common development library, as is implied by our diagram, or it can be individual programmer libraries.

You can use one set of libraries all the time, or you can have project-based libraries (that is, libraries that are used only for the duration of a specific project).

You can also define secured libraries (not shown), perhaps for vendor source, from which source can be checked out, but into which changes cannot be promoted.

You can define multiple data libraries (called "Explode" libraries, not shown) into which data objects can be duplicated for testing file changes or supporting multiple divisions with different data.

You can define alternate promotion paths (not shown) to support objects that need to be promoted to a different set of libraries or paths.

In the diagram, each pair of "from" and "to" libraries is defined within the application definition, and is referred to as an *application level*. The DEVTEST and ACPTTEST libraries are the From/To library pair for application AP level 1. The ACPTTEST to PRODTEST libraries are the From/To library pair for application AP level 2. Checkout rules, promotion rules, and other defaults are set for the entire *application definition*, but they can be overridden when you add application levels. The application definition contains rules that have been set in response to questions as:

- Should source be archived when it is replaced? Where?
- Should your application definition enforce object naming rules?
- How do you want source code and objects handled during a promotion?
- Do you want to distribute objects to other computers when you promote changes? Objects and source? Source only?
- Should someone in your organization approve changes before they are implemented?
- What user profile should own the objects TURNOVER® Lifecycle Manager creates?
- What object types does your software application contain?

These are just examples of some of the rules contained in a TURNOVER® Lifecycle Manager application definition.

Some of the answers may differ depending upon which application level you are modeling. The list of people who need to approve a change may differ at each level, and you will more likely distribute changes when you promote to production than when you promote to acceptance test.

Even if you do not have authority to define applications, you can view them.

---

## Workflow

Once you have been assigned a task and have determined what objects you need to change or create, you are likely to take the following actions.

### Check out objects and source from production

TURNOVER® Lifecycle Manager copies the source from your production source library to your development library (non-source objects are duplicated).

Depending on how your application is set up, TURNOVER® Lifecycle Manager might not allow others to check out the same objects. Using the TURNOVER® Lifecycle Manager cross-reference facilities, you can identify dependent objects, so you can feel confident that you know about all the objects you need to change or re-compile.

### Develop objects in Test

You develop new objects, change existing objects, and test objects in a common development library or in your own development library.

### Create a TURNOVER® Lifecycle Manager form

When you are ready to promote source and objects, you build a TURNOVER® Lifecycle Manager form. A *form* is a promotion request containing all of the information TURNOVER® Lifecycle Manager needs to promote to a particular level. It can also contain comments, lock checks, create-command-overrides, and commands such as Override Data Base File (OVRDBF) to be run when objects are promoted.

### Pre-run and error-check form

Before you run your form, you can error-check it to minimize the likelihood of problems later.

### Obtain form approval

You notify all of the people on the approval list, by system message or by email. Each user reviews the form, and then approves it.

### Run TURNOVER® Lifecycle Manager form

You (or the TURNOVER® Lifecycle Manager Administrator) submit the form to run, either immediately or when you specify.

### Form processing

TURNOVER® Lifecycle Manager copies the source to the target library and compiles the object into the appropriate target object libraries.

TURNOVER® Lifecycle Manager copies or moves non-source objects, as your application definition indicates, and duplicates source that you do not want compiled, such as RPG /COPY code or S/36 OCL. Old source and objects can be archived. TURNOVER® Lifecycle Manager handles the source and objects in the From library as your application indicates. If errors occur, TURNOVER® Lifecycle Manager rolls your changes back.

### **Check out from Acceptance (or any intermediate application level)**

If errors are detected during acceptance testing, you check those objects out of the acceptance test level; then you revise them and promote them back into Acceptance for additional testing. This might happen several times.

### **Copy the form**

When acceptance testing is complete, you can copy your original form to create a new form that promotes objects into production. All library references and similar defaults are translated when you copy a form. A TURNOVER® Lifecycle Manager Administrator typically performs this and the following steps.

### **Run Production form**

The Administrator submits the form to run. It will be rolled back if errors are detected.

### **Back out form**

If you are not satisfied with the result of running a form, you can roll it back. You can roll back the entire form, or any subset of the objects that are on it.

### **Distribute form**

A form is also a distribution request. After the form runs successfully on the development computer, TURNOVER® Lifecycle Manager can distribute it to other computers in the network, either immediately, or at some later date and time.

TURNOVER® Lifecycle Manager can distribute both native IBM i objects, and objects that are in files systems on other platforms.

### **Receive and run form**

On your remote production computers, users can run the forms just as they were run on the development computer. If you distribute from level 2 of your AP application definition, you need a corresponding application definition for AP level 2 on the remote computer. That application definition controls how the objects are managed on *that* computer.

It is typical to distribute only objects; therefore the promotion method specified by the remote application definition is usually *MO (Move Object)*. However, you can configure TURNOVER® Lifecycle Manager to send either source alone or source and objects, and compile objects on the target computer.



### Update form status

When objects have been distributed and installed on each remote computer, TURNOVER® Lifecycle Manager updates the status on the development computer and notifies people responsible for the change.

### More information

For more information, see the *TURNOVER® Lifecycle Manager Developer's Guide*.

---

## Creating application definitions

To manage all your objects, you might need only a few applications, or you might need thousands. Some users define several, each represented by a different set of libraries. Generally, you should define one application for each of your production program libraries. If all of your program objects are in one or just a few libraries, you can decide to define multiple applications anyway. You can define the applications to share the same production (target) libraries.

When deciding how to define your applications, consider these questions:

- Are some of your applications written in house and some supplied by a vendor?
- Are object naming standards different for each of your applications?
- Do different groups of programmers work on different applications?
- Do different people approve changes to different applications?
- Are changes for one application being distributed to one computer and changes for other applications distributed to other computers? (For example, accounting software might be in use on one computer, order entry software on another.)

There are two ways to create application definitions:

- copy one of the models that are provided with TURNOVER® Lifecycle Manager, or
- build your own application definitions.

### Recommendation

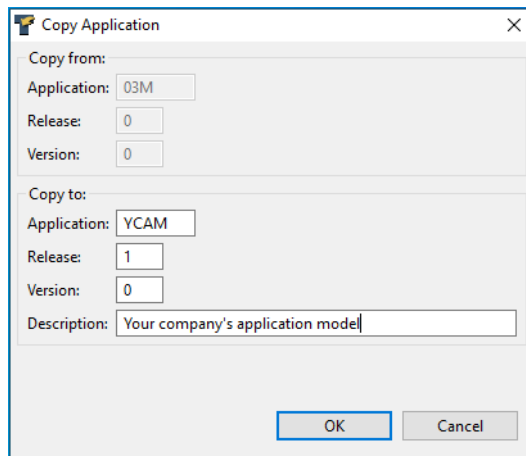
Implement one application definition, and then test it fully, before implementing others.

---

## Copying an application model

- 1 In the TURNOVER® Client, navigate to Applications, right-click a model, and then click **Copy Application**.

The Copy Application dialog box appears.

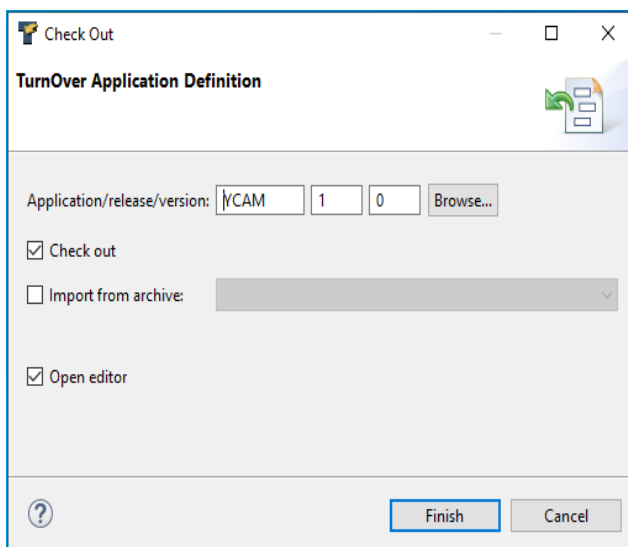


- 2 Specify an application name, release, version, and description information, and then click **OK**.

TURNOVER® Lifecycle Manager asks you want to check out the new application definition.

- 3 Click **Yes**.

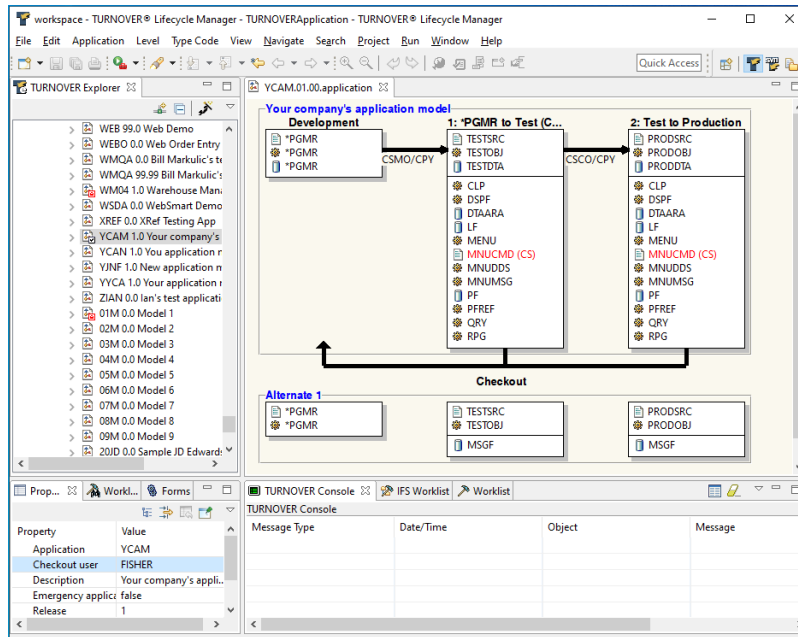
The Check Out dialog box appears.



- 4 Click **Finish**.

The new application definition opens in the Application Designer.

Your application definition is created, checked out, and opened in the editing pane of the TURNOVER® Lifecycle Manager Client application window.



- 5 You can now edit the custom model to make it match your development requirements. For example, you might:
  - add explode locations to one or more type codes, to propagate changes into multiple test locations during promotions
  - add alternate promotion paths, for objects that need to follow a slightly-different-than-default path into production
  - add additional object type codes for other languages or platforms
  - configure distribution at one or more levels.
- 6 Test your model. When you are satisfied with it, copy it, and then customize it for a particular software application.

## Using the New Application Definition wizard

- 1 In the TURNOVER® Lifecycle Manager Client application, right-click the Applications subsystem, and then click **New Application**.

The New Application Definition wizard appears.

The screenshot shows the 'New Application Definition' wizard at the 'Description' step. The title bar says 'New Application Definition'. Below the title bar, the section is 'Description' with the instruction 'Name your application and identify its purpose.' The form contains the following fields and options:

- Application:** A text input field.
- Release:** A numeric input field with '0' entered.
- Version:** A numeric input field with '0' entered.
- Description:** A text input field.
- Emergency application:** An unchecked checkbox.
- LANSa application:** An unchecked checkbox.
- Structure:**
  - Number of QA/Testing environments (excluding development):** A dropdown menu with '1' selected.
  - Number of permanently-locked environments:** A dropdown menu with '0' selected.

At the bottom, there are navigation buttons: '< Back', 'Next >', 'Finish', and 'Cancel'. A help icon (?) is also present.

- 2 Specify the name and structure of your application definition.
- 3 If the application will be used for emergency changes, select **Emergency application**.

If your software application includes vendor-supplied source and objects that must be preserved, or if you are managing multiple versions of the same software product, you might need at least one permanently locked level. Permanently locked levels reside above production. Items can be checked out of permanently locked levels, but changes are never promoted back into them.

- 4 Click **Next**.

The Development Libraries and Path page appears.

The screenshot shows the 'New Application Definition' wizard at the 'Development Libraries and Path' step. The title bar says 'New Application Definition'. Below the title bar, the section is 'Development Libraries and Path' with the instruction 'Tell TURNOVER where your programmers will be making their software changes.' The form contains the following fields:

- Development Environment:**
  - Environment description:** A dropdown menu with 'Development' selected.
  - Source library:** A text input field with '\*PGMR' entered.
  - Source path:** A text input field.
  - Object library:** A text input field with '\*PGMR' entered.
  - Data library:** A text input field with '\*PGMR' entered.
  - IFS path:** A text input field with '\*PGMR' entered.

At the bottom, there are navigation buttons: '< Back', 'Next >', 'Finish', and 'Cancel'. A help icon (?) is also present.

- 5 Specify the your development source, object, and data libraries and the path for IFS objects.

The default value, \*PGMR, tells TURNOVER® Lifecycle Manager to retrieve this information from the user's Programmer information.

- 6 Click **Next**.

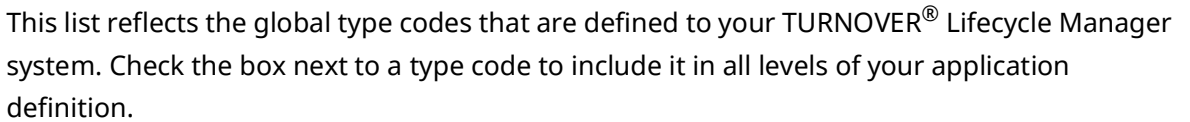
The QA Libraries and Path page appears.

- 7 Specify the source, object, and data libraries and IFS path belonging to your first intermediate QA or Testing level.
- 8 Specify a description for this level, and then click **Next**.
- 9 Repeat the previous step for each intermediate level you specified on the Description page.
- 10 On the Production Libraries and Path page, identify the source, object, and data libraries and the IFS path that belong to your production environment.

- 11 Specify the description, and then click **Next**.

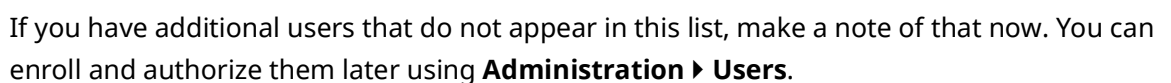
The Type Codes page appears.

- 12 Specify the object types that make up your software application.

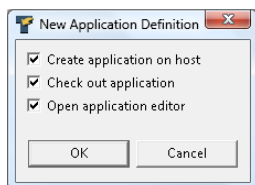


13 Click **Next**.

14 Select the users who should have authority to this application definition.



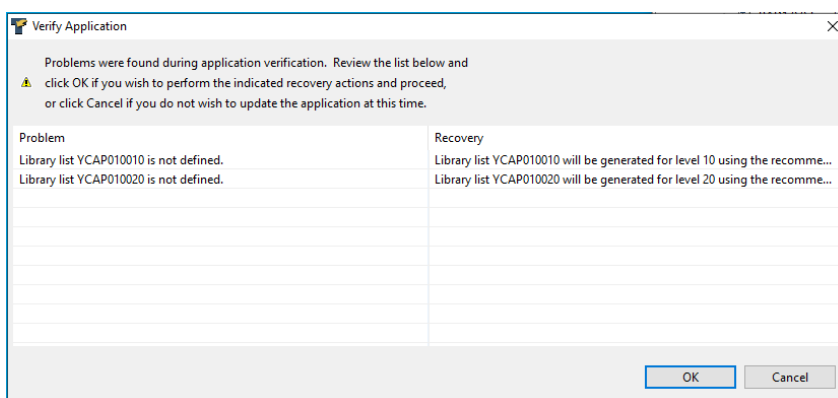
The New Application Definition dialog box appears.



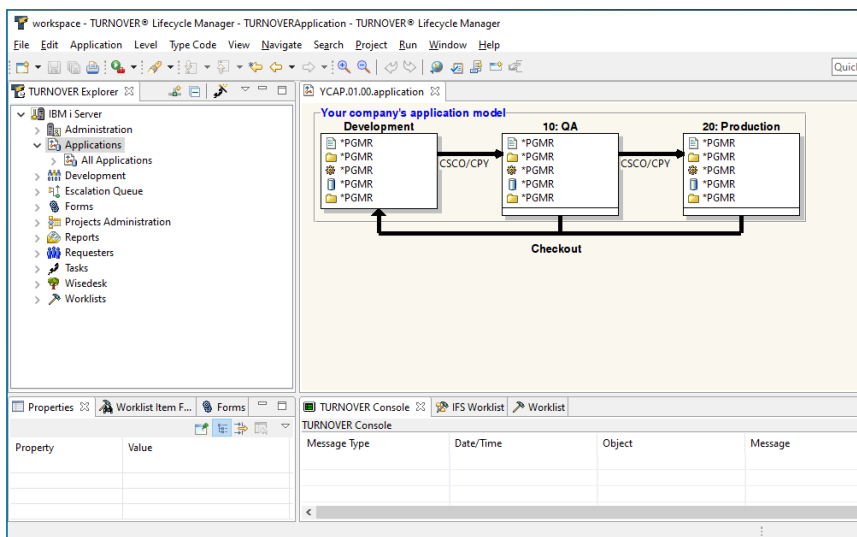
16 Make sure that all the options are selected, and then click **OK**.

TURNOVER® Lifecycle Manager creates the application definition, checks it out to you, and then opens it in the Application Designer.

It also verifies your application definition. If any items do not exist, such as library list objects, TURNOVER® Lifecycle Manager asks if you want to create them.



17 Click **OK**.



18 If you need to add any type codes and users, use the Administration wizards to create them now.

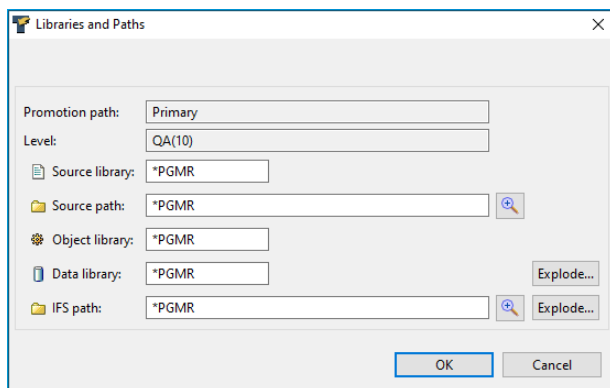
- After enrolling new users, you can manage their application authorities in their User Properties.
- To add a type code, return to the Application Designer, right-click a level, and then click **Add type codes**. TURNOVER® Lifecycle Manager adds the type code to all levels.

## Introduction to the Application Designer

The Application Designer enables you to visualize your application definition as you create it.

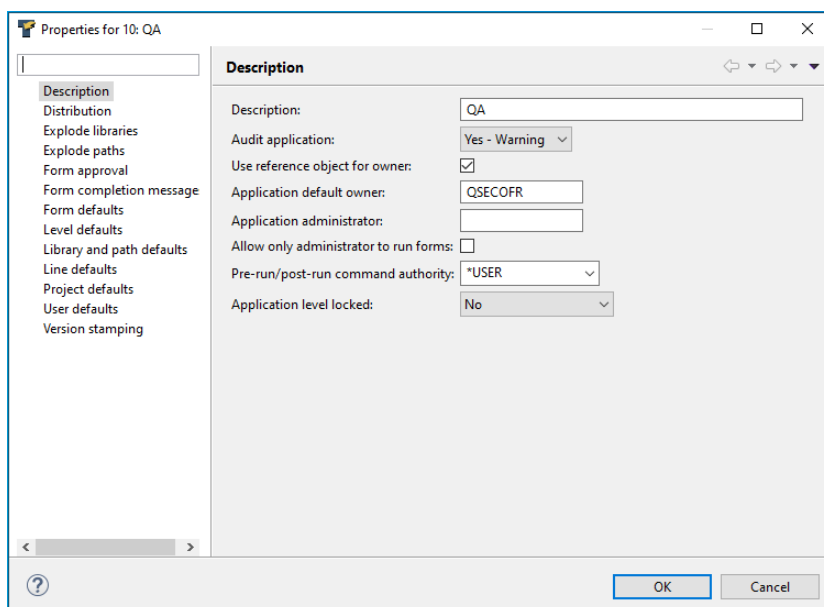
### Level libraries and paths

To manage the libraries and paths specified for an application level, right-click any application level, and then click **Change Libraries and Paths**.



### Level properties

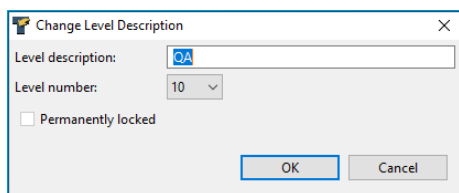
To manage the properties of an application level, right-click the level in the application image, and then click **Level properties**.



### Level description

An application level's description comprises text, a level number, and a lock indicator. To display the level description, right-click the level, and then click **Change level description**.





- Do not change the level number when the application is in use.
- Intermediate levels of an application definition cannot be permanently locked.

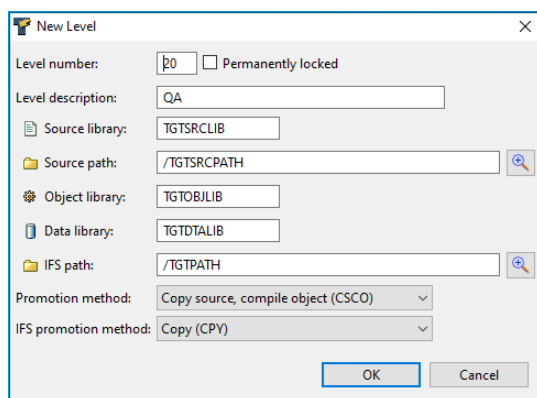
## Inserting application levels

When you create an application definition using the New Application wizard, TURNOVER® Lifecycle Manager leaves space between levels so that you can insert new levels later.

### To insert an application level

- 1 Right-click a level, and then click either **Insert level before** or **Insert level after**.

The New Level dialog box appears.



- 2 Specify the details of the level, and then click **OK**.

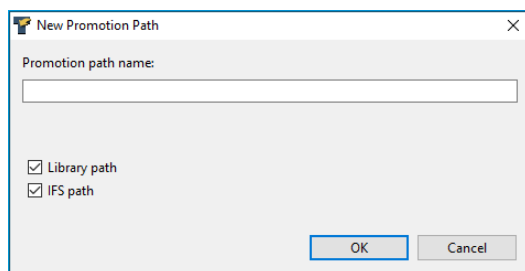
TURNOVER® Lifecycle Manager warns you if inserting the specified level number will cause higher levels to be re-numbered, so you can change it if necessary.

## Promotion paths

To store a set of objects in a different location than other objects, create a *promotion path*.

### To create a new promotion path

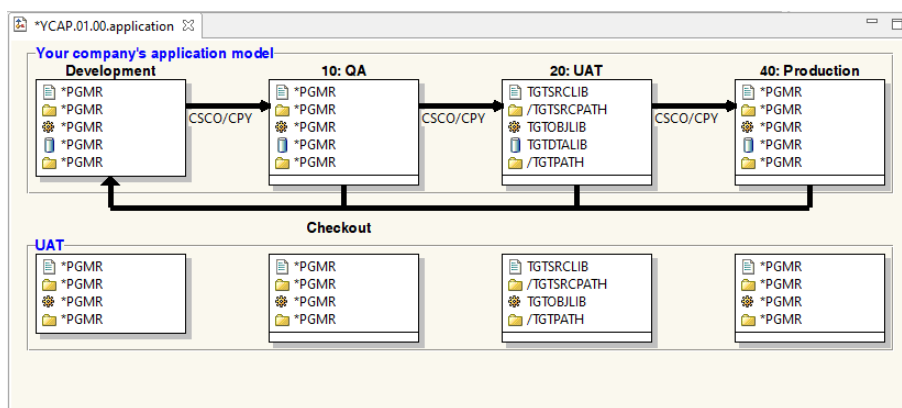
- 1 Right-click anywhere in the application diagram, and then click **New promotion path**.



- 2 Specify a name for the promotion path, and then indicate whether it has libraries or IFS paths (or both) associated with it. Once created, you can change the libraries or paths associated with the promotion path, and then assign type codes to it.

### Example

A promotion path, when added to your application diagram and populated with some type codes, looks something like this:

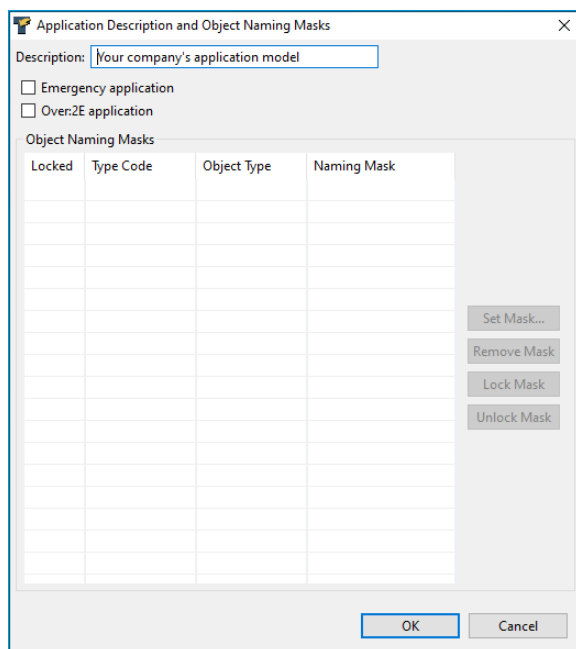


## Application description and object naming masks

You can force object names to match a naming standard. This is also where you maintain the primary text description for your application definition, as well as whether it is used for emergency changes.

### To access the application description

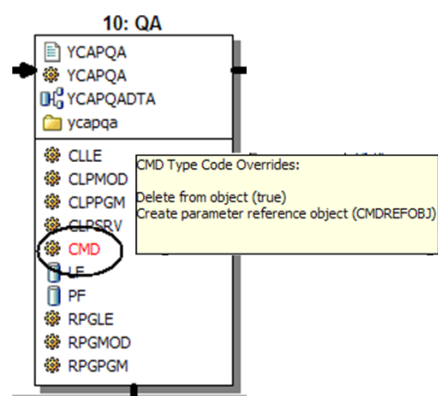
Right click anywhere in your application diagram, and then click **Application description and object naming masks**.



## Type code overrides

To customize a type code at the application level, double-click it.

After you have made your changes, the type code name changes color. To see the customized values, move the pointer over the type code.



### To manage the colors used

- 1 On the Window menu, click **Preferences**.
- 2 Navigate to TURNOVER® Lifecycle Manager > Administrator, and then go to the Application Designer page.

## 3 Configuring auditing needs

TURNOVER® Lifecycle Manager ensures that all software changes are completely traceable, from the initial request to the final deployment of objects.

---

### Define a software development process

A well-defined software development process must include a basic project management scheme. You need to be able to process and track issues, defects, and enhancement requests from the initial source all the way through to completion of the associated development tasks.

TURNOVER® Lifecycle Manager provides functionality to do just that. TURNOVER® Lifecycle Manager enables users to enter issues into the Helpdesk projects. If it is determined that an issue requires a programming change, the project administrator can assign the issue to a task in a *working* project. The developer picks up the task, reviews the details to understand the required change, and creates a worklist. The developer can add affected objects to the worklist, include them on a form for promotion to a Test level, and finally back up to Production. You can use escalation processing to notify designated users by email at every step in the software life cycle process.

---

### Application settings

The following components are all controlled by the application definition settings (TURNOVER® Lifecycle Manager main menu option 1, Work with Application Definitions, or the Applications option in the TURNOVER® Client Administrator module).

#### Audit your application

Set **Audit application** at the Production level of the application to either **W** (warning) or **E** (error). This setting checks if an object or source member has been changed outside of TURNOVER® Lifecycle Manager when you are checking out an item or running a form.

- **W** displays a warning message, but you can still run the form.
- **E** prevents you from running the form.

## Archiving source and objects at production level

Turn on source as well as object archiving (controlled separately) at the Production level of the application definition.

Whenever source and objects are replaced by modified versions, TURNOVER® Lifecycle Manager stores the old versions for subsequent access. It can then use the archived objects if it becomes necessary to recover or roll back recently deployed objects. You can access and compare the old source to the new source for auditing purposes, or to understand the change that was made. You can periodically save archive source and objects to tape and store them in a secure location.

## Form approval lists

Define an approval list at the highest level of the applications so that the appropriate users can *sign off* a change before it is promoted to Production

Also consider including approval lists at other levels of the application where it is important to control the installation of changes.

The approval list can include one or more users or groups of users. You can either require each user to approve the form or to allow selected users to approve for all others. When you define an approval list for an application, forms at that level can run only when approval has been received. If the contents of a form have been changed *after* approval has been granted, the form requires re-approval.

---

## Versioning and object stamping

You can activate object versioning and object stamping, for items promoted by means of a TURNOVER® Lifecycle Manager form

To view this object level information, run the TURNOVER® Lifecycle Manager TDSPOBJD command.

```
MM/DD/YY 11:22:33      Display Object Description - TURNOVER®      SLS IN-HOUSE
Object Information:
Object . . . . . : MENRUSRL      Object version . . . . . : 036.000
Library . . . . . : SOFTMENU61    Appl/Rel/Ver . . . . . : SM61/00/00
Type . . . . . : *PGM
Object owner . . . . . : SOFTMENU61
Creation Date/Time . . . : 10/30/12 12:56:05
Text . . . . . : List Users Program
Source Information:
Source file . . . . . : QRPGLSRC      F2=View Source
Library . . . . . : SM61SRC          F10=Compare with Archive
Member . . . . . : MENRUSRL
Attribute . . . . . : RPGLE
Source file date/time . . : 9/13/12 9:46:52
TURNOVER Information:
Created by form . . . . . : 0141707      New special value to list just Option Gr
Project/task reference . . : SE610037     New special value to list just Option Gr
Type code . . . . . : RPGLE

F3=Exit F12=Cancel F15=Checkout History F18=Object History      Bottom
F5=*FULL F6=View form F7=DSPPGM F8=*SERVICE F9=View task
```

The panel includes standard IBM i data, such as object and source information, and TURNOVER® Lifecycle Manager information. You can find:

- the form that was used to promote it, the project/task reference, and the associated type code
- the details that initiated the change
- the associated Helpdesk task that defined the initial change request
- the approval list to determine who signed off on the promotion of this object
- what other objects were promoted along with the subject object and if any special commands or programs were also run as part of the promotion
- other historical information, such as checkout history and object change history.

---

## Defining appropriate authorities

The scope of activities that any user can perform in TURNOVER® Lifecycle Manager is defined by their authority settings.

These authority settings, which allow users to create and run promotions, can be defined by the TURNOVER® Lifecycle Manager Administrator at a global level across all applications or on an application-by-application basis. The administrator can also define authorities at either a user or group profile level, or at both levels. User authority is then the aggregate of the individual user and associated group profile.

Developers should be able to create and possibly run forms up to your Test or Quality Assurance level, but a developer should not usually be authorized to run their own changes up into Production. You can implement this concept by removing form run authority at the Production level of your application.

Limit access to only those applications that a user needs to do their job.

---

## Audit a change

You can use object history, forms, and reports to help you trace a change from beginning to end. This information will likely be of interest to your auditors.

### Object history

To review the promotion of a particular object, from initial request to ultimate deployment, use the object history panel (main menu option 4, Work with Object History). On this panel, you can:

- find out the last time an object was changed
- get a list of all changes that were made to an object in TURNOVER® Lifecycle Manager
- view archive information, distribution, and checkout history
- browse or restore archived source

- perform a source compare between the current and archived versions of the source
- determine on what form the object was promoted, and see the associated task under which the work was performed.

If you have been using the Project system, the working task always has a link back to the originating Helpdesk task that triggered the change in the first place.

## **TURNOVER® Lifecycle Manager forms**

In TURNOVER® Lifecycle Manager, the form is used to promote objects from one environment to another. Once a form has run, it can be viewed to provide details about the promotion. You can view the stamped information or go into Object History to determine the form number associated with the promotion of a particular object. You can get the following information by viewing the form (main menu option 6, Work with Forms):

- other items that were promoted with the object of interest
- pre- or post-run commands that were run as part of the promotion process.
- how the objects were promoted – were they *moved* to the target environment or *compiled* there?
- the users that approved the form or signed off on the change
- the task under which the object was promoted
- when the form ran and the user that ran it
- other systems to which the objects were distributed
- the name of the archive library that contains the replaced objects.

---

## **TURNOVER® Lifecycle Manager reports**

TURNOVER® Lifecycle Manager provides a number of reports (main menu option 10, Reporting menu). You can also generate customized reports.

## 4 Change process

This section describes how the TURNOVER® Lifecycle Manager pre-edit and promotion processes work, and explains some of the rules and exception conditions TURNOVER® Lifecycle Manager looks for.

It is convenient to think about the change process by considering these generic object types:

- source-only “objects”
- source-compiled objects
- non-source objects
- non-IBM i types (such as CASE tool interface types).

Generally, all objects in each group are handled in the same way, but there are exceptions: for example, physical files with associated logical files, and message files.

The main variables that control the change process are:

| Variable                              | Description                                                                |
|---------------------------------------|----------------------------------------------------------------------------|
| Method                                | CSCO, CSMO, CSCD, CS, CD, MO, COPR, *NULL                                  |
| Action code                           | A=Add, R=Replace, D=Delete                                                 |
| TURNOVER® Lifecycle Manager Type code | Found in the TURNOVER® Lifecycle Manager type code file of the application |
| Object type attribute code            | RPG, CLP, PF, LF, and so on                                                |
| Create Command                        | Found in the TURNOVER® Lifecycle Manager global type code file             |

See also ‘Working with Type Code Definitions’ in the *TURNOVER® Lifecycle Manager User Guide*.

A promotion job consists of four phases:

- 1 Pre-run processing
- 2 Promotion processing
- 3 Post-run processing



#### 4 Finalization.

The finalization phase may be performed asynchronously (that is, in another low-priority job) to minimize the impact on your users and to make objects available to users more quickly. This is controlled by the **Submit finalization** application definition parameter.

You can select several forms to run together as a group.

Whether you are running one form or a group of forms, the process runs under one job, named `TOnnnnnnnG`, where `nnnnnnn` is the form number of the lowest form number in the group. All objects on all of the forms are promoted before running the post-run and finalization phases for any form. Once the post-run phase is completed (and before starting finalization) the form status for all forms is set to `FIN-OK` (or a variation of this), indicating that the user can use the objects that were promoted and that the finalization phase has not been run or has not completed yet. If the finalization phase is submitted, the job is named `TOnnnnnnnnF`.

---

## Form promotion processing flow

### TURNOVER® Lifecycle Manager forms selected to run

All forms are set to `IN-USE`. Batch job named `TOnnnnnnnnG` is submitted to process forms.

### TURNOVER® Lifecycle Manager form error-checking (all forms)

Most of these checks are performed when the form is submitted or during the error-checking process. This process is usually done in batch. If only one form is run, you can error-check interactively.

- 1 Check user's authority to submit a form.
- 2 Look in the checkout file for missing objects. The form's **Project task reference** code is used to look for objects checked out with the same reference that are not included on the form.
- 3 Check to see if the test member and object match. Object description date and times are used to ensure a match (where Method is CSCD or CSMO only).
- 4 Check to see if test members or objects have changed since form was approved.
- 5 Check to see if the production member exists (must exist, if replace or delete; can exist if add and will be archived).
- 6 Check to see if production object exists (must exist, if replace or delete; must not exist if add).
- 7 Check program references. Will the required files be found in the run-time library list?
- 8 Check to see if the source member had been checked out.
- 9 Check the programmer.
- 10 Check the object name against the naming mask (new objects only).

- 11 Check to see if the reference (REF\*) objects exist.
- 12 Check to see if the from and target source files are the same length (S/36 environment only).
- 13 Check to see if the same object name, target library and type is on more than one form (except if first occurrence is a delete).
- 14 If any form has errors, TURNOVER® Lifecycle Manager produces an error log and returns all forms to READY status. No further processing occurs.
- 15 Call the Pre-Run Error Check Exit program (Exit 13).
- 16 If no errors exist, call the Pre-Form Group Exit program (Exit 19).
- 17 Run any TestBench \*PRERUN tests before any objects are promoted. Form will remain in IN-USE status during this phase of processing.

### **TURNOVER® Lifecycle Manager form pre-run processing (Phase 1)**

- 1 Set the form status to RUNNING.
- 2 Check the form dependency. If dependent form failed or has not been run, terminate job.
- 3 Check the date-time range specified in form header.
- 4 Check for object locks and allocate, if specified.
- 5 Call the interface programs, if appropriate.
- 6 Call the Form Initialization Phase Exit program (Exit 8), if specified.
- 7 Run the pre-run commands specified as \*PREFORM. Overrides associated with one form will *not* be in effect for other forms.

### **TURNOVER® Lifecycle Manager promotion processing (Phase 2)**

- 1 Promote IFS objects with a \*BEFORE sequence in the global type code.
- 2 Create archive library TnnnnnnnnDy, where nnnnnnnn=form number and y=ASP. (Multiple libraries may be created if applications use multiple ASPs).
- 3 End physical and logical (access path) file journaling.
- 4 Remove any pre-existing triggers and/or constraints.
- 5 Copy source and move objects to TnnnnnnnnD libraries. (If program type object and creation command parameter REPLACE is \*YES, TURNOVER® Lifecycle Manager will create duplicate of production object into TnnnnnnnnD archive library).
- 6 Copy the source to the target source files and libraries. If method is COPR, source is not copied to the target library.
- 7 Run pre-run commands associated with the line if specified as \*PRELINE, using library list \*CURRENT or \*APPL, depending on form command definition.

- 8 Replace the library list with application library list.
- 9 One of the following:
  - Compile object into target library using target source.
  - Move test object to target library.
  - Duplicate test object into target library.
- 10 Run the post-run commands associated with the line if specified as \*POSTLINE, using library list \*CURRENT or \*APPL, depending on form command definition.
- 11 Change the creation parameters.
- 12 If physical or logical file, add members.
- 13 If physical files, copy data from old file into each target file member.
- 14 Promote IFS objects with an \*AFTER sequence in the global type code.

### No errors detected

If none of the pre-run or mainline steps cause the form to end, then the next form is set to **RUNNING**, and these steps are repeated on that form. Status of the current form remains **RUNNING** after these steps have completed.

### Error detected

If any errors occur during Phase 2 or 2, the form that had the error is recovered according to the **Recovery option** specified for that form. The error-threshold is also determined by the form that has the error, and the error count is initialized to zero at the start of each new form. All previously processed forms are also recovered based on the recovery option of the form that had the error.

### TURNOVER® Lifecycle Manager post-run processing (Phase 3)

(All Phase 2 is completed for all objects and forms before Phase 3 begins.)

- 1 Set the owner of the new object.
- 2 Set the authority of the new object.
- 3 Restart file journaling, if appropriate.
- 4 Replace pre-existing triggers and/or constraints.
- 5 Explode the data objects (repeated for all data type objects in each explode library).
  - Create the data explode archive library, TnnnnnnnnE (once).
  - Move the data object into archive library. Create alias name, if duplicate.
  - Duplicate the physical files from the primary target library into the explode library.
  - Create the members of physical files in explode library and copy data.

- Create the logical files in explode library.
  - Set Create parameters of the logical files.
  - Add members to the logical files.
  - Set authority and ownership of the explode library objects.
- 6 Call the interface post-run programs.
  - 7 Call the Form Finalization Processing Exit program (Exit 1), if appropriate.
  - 8 Call the post-run commands specified as \*POSTFORM.
  - 9 Set the form status as follows:

| Status  | Description                                                                                       |
|---------|---------------------------------------------------------------------------------------------------|
| FIN-WRN | Finalization phase, warning messages generated during processing                                  |
| FIN-ERR | Finalization phase, error messages generated during processing (one or more objects not promoted) |
| FIN-OK  | Finalization phase, no errors detected                                                            |

- 10 If auto-distribution and status is FIN-OK for all forms in the group, submit the distribution job.
  - Status changes to FIN-PND (distribution pending) if finalization has not yet completed, or DIST-PND if finalization has completed.
  - After distribution completes, the status changes to FIN-DIST, FIN-PART, DISTRIB, or PARTIAL.
  - When confirmation is received, the status changes to FIN-CONF or CONFIRM. Any FIN-status is updated at the end of the finalization job. If distribution is automatic, the forms are distributed as a form group. If the remote computer is set up to run forms automatically, they run as a form group when they are received.

When Phase 3 has completed for all forms in a group, the Finalization processing Phase 4 processing will start.

## Finalization processing

Steps 1-6 are done in a separate finalization job if **Submit finalization** is set to Y.

- 1 Archive the source.
- 2 Delete the test source member.
- 3 Delete the test object.
- 4 Rebuild cross-referencing.

- 5 Delete the `TnnnnnnnnD` libraries and `TnnnnnnnnE` libraries, if **Archive Objects** is N and no errors were detected.
- 6 Set the form status to `RAN-OK`, `RAN-WRN`, or `RAN-ERR`.
- 7 Send the completion messages to host computer (if a distributed form).
- 8 Send the form completion messages.
- 9 Call the Form Completion Processing Exit program (Exit 11).
- 10 Run any TestBench `*POSTRUN` tests that have been specified.
- 11 Call the Post Form Group Exit program (Exit 20).
- 12 Print TURNOVER® Lifecycle Manager logs.

Most of these steps are conditional and depend on the major control variables (see *'Change process' on page 52*).

---

## Rolling back a form

If a TURNOVER® Lifecycle Manager form fails and you specified that TURNOVER® Lifecycle Manager should roll back the changes, TURNOVER® Lifecycle Manager reverses the change process, copying and moving source and object back from the target libraries to the from libraries and from the archive libraries to the target libraries.

You can also roll back a form after it has completed successfully by creating and running a recovery form. This procedure requires that objects are being archived and that they are on-line at the time you run the recovery form. When you create a recovery form, you can specify that it contain only items that were promoted; this saves you from having to edit and remove other items after you create the form. Also, if any physical files on the form have been updated since the form ran, be aware that the data will be copied from the new file to the old file using the copy command parameter `CPYFMT` of `*MAP *DROP`. See the "Creating a recovery form" in the *TURNOVER® Lifecycle Manager User Guide*.

---

## Rerunning a failed form

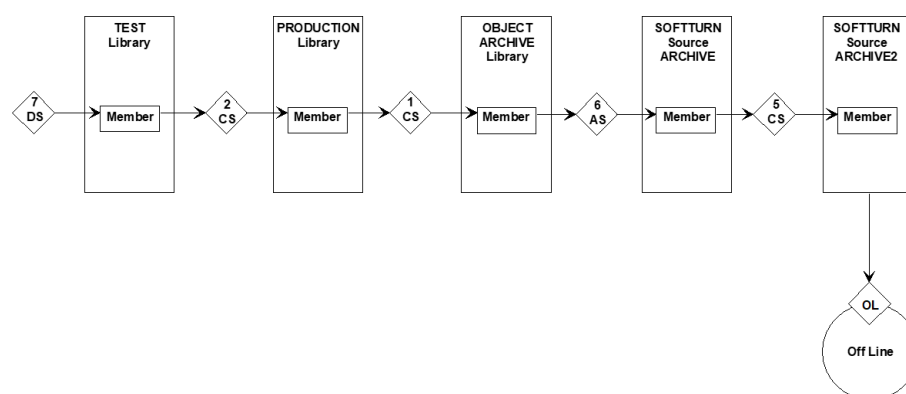
If a form fails because someone cancelled the job, your power or computer failed while the form was running, or for any other reason where the form status is `RUNNING`, even if the job is not running, you can restart the form. You do so by selecting the form with option 8=Work w/ job on the Work with Forms panel and then, on the Work with Form Group Jobs panel, selecting the form's job with option 7. When you restart the form, its status must be set to `RUNNING`.

If the form promotion job had run to completion and the form status is `RAN-ERR`, you must copy the form and rerun it or create a recovery form to back-out the items that were promoted (see previous section). When you copy a form with status of `RAN-ERR` to the same level, TURNOVER® Lifecycle Manager omits lines with a line status `RAN-OK` or `RAN-WRN`; all other items will be included on the new form. This eliminates the need to edit a form to remove items that promoted successfully.

**Important** If you are distributing changes to production (target) computers and one or more of these are operating at an earlier IBM i Release level, objects must be compiled with the `TGTRLS` parameter, using a value of `*PRV` or a specific version release designation such as `V4R1M0`. You must set both the `TGTRLS` parameter of the create commands in the TURNOVER® Lifecycle Manager object type file, *and* the TURNOVER® Lifecycle Manager global `TGTRLS` parameter. See Utility Menu options 4 and 5.

## Source-only objects

Source-only objects, such as help text and copybook code, can be promoted to production using only one TURNOVER® Lifecycle Manager method, `CS` (Copy Source).



Some steps are conditional. The Object Archive library is saved only if the application is set up to archive objects. You initiate the step to save the Object Archive library some time after the TURNOVER® Lifecycle Manager job runs. Source archiving depends on application defaults as well.

CS=Copy Source

MO=Move Object

CD=Create Duplicate Object

CO=Compile or Create Object

AS=Compress and Archive Source

DS=Delete Source

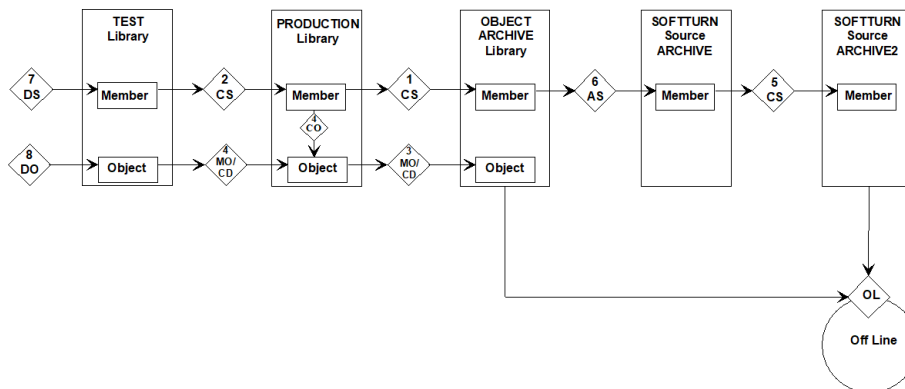
DO=Delete Object

OL=Off-Line Save

## Source-compiled objects

The diagram below describes the change process when you specify TURNOVER® Lifecycle Manager Method **CSCO** (Copy Source, Compile Object), the most frequently used change method. This also describes **CSMO** (Copy Source, Move Object) and **CSCD** (Copy Source, Create Duplicate Object), if you substitute **MO** or **CD** for **CO** in this diagram (Step 4).

If you use **CSMO** or **CSCD**, TURNOVER® Lifecycle Manager makes sure that the source and object match before allowing you to submit the form. It also checks to see if the source or object has changed since it was submitted, to prevent tampering with or substitution of another source member or object after job submission.



Some steps are conditional. The Object Archive library is saved only if the application is set up to archive objects. You initiate the step to save the Object Archive library some time after the TURNOVER® Lifecycle Manager job runs. Source archiving depends on application defaults as well.

CS=Copy Source

MO=Move Object

CD=Create Duplicate Object

CO=Compile or Create Object

AS=Compress and Archive Source

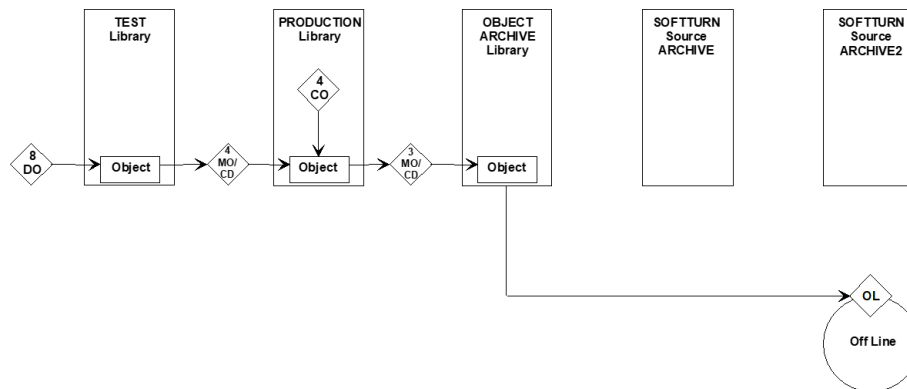
DS=Delete Source

DO=Delete Object

OL=Off-Line Save

## Non-source objects

The diagram below describes the change process when you specify TURNOVER® Lifecycle Manager Method MO or CD, where source does not exist or is not being used at the time the object is promoted to production. This method is used most frequently on a production computer that is the recipient of distributed changes, but also applies to some object types on a development computer such as data areas, data queues, output queues and job descriptions.



Some steps are conditional. The Object Archive library is saved only if the application is set up to archive objects. You initiate the step to save the Object Archive library some time after the TURNOVER® Lifecycle Manager job runs. Source archiving depends on application defaults as well.

CS=Copy Source

MO=Move Object

CD=Create Duplicate Object

CO=Compile or Create Object

AS=Compress and Archive Source

DS=Delete Source

DO=Delete Object

OL=Off-Line Save

## Additional rules and conditions

### TURNOVER® Lifecycle Manager methods

The method you specify in your application line defaults controls how TURNOVER® Lifecycle Manager transfers source or objects (or both) from test to production.



| Method |                                          | Description                                                                                                                                                                                                                                                           |
|--------|------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CSCO   | Copy Source, Compile Object              | TURNOVER® Lifecycle Manager copies the source member to the target library and compiles it.                                                                                                                                                                           |
| CSMO   | Copy Source, Move Object                 | TURNOVER® Lifecycle Manager copies source to the target library and moves the test object to production, replacing the production source and object. Source and object must match before being moved.                                                                 |
| CSCD   | Copy Source, Create Duplicate Object     | TURNOVER® Lifecycle Manager copies source to production and creates a duplicate of the test object in production, replacing the production source and object. Source and object must match or an error is generated.                                                  |
| CS     | Copy Source-only Objects                 | TURNOVER® Lifecycle Manager copies source-only objects to production.                                                                                                                                                                                                 |
| MO     | Move Object                              | TURNOVER® Lifecycle Manager moves the object from test to production. No source is transferred.                                                                                                                                                                       |
| CD     | Create Duplicate Object                  | TURNOVER® Lifecycle Manager creates a duplicate object in production from the object in test. No source is transferred.                                                                                                                                               |
| COPR   | Compile or Create Object from Production | Use this if you need to recompile objects from source in the target library or a higher-level library (Compile in-place). If the object is a non-source object type, it means create the object from the command in the TURNOVER® Lifecycle Manager Object Type file. |
| *NULL  | No action                                | Do not move, duplicate or create objects. This may be appropriate where you only want to distribute and install changes on a remote computer, not recreate them on the development computer.                                                                          |

## CREATE parameter and initial object authority rules

When creating or moving an object to production, TURNOVER® Lifecycle Manager references other objects in production or test; or it uses system defaults or a reference object you specify, to determine what initial parameter settings to use for the new object and what object authority to grant to the new object. All parameters that you can change using a change command can be set using a reference object.

## Create parameters

| Parameter | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| P         | Use the production object as a reference for relevant parameters when creating this object. If the object is being added, use the named reference object (REF* or equivalent), if present. If the reference object is specified but is not found, an error will be issued. If the Reference object is not specified (that is, is blank), TURNOVER® Lifecycle Manager uses the create command defaults, as specified on the TURNOVER® Lifecycle Manager Object Type File command field as if this field had been coded S. If the application is a multi-level application, and Create Parameters is P and a production object exists at a higher level, TURNOVER® Lifecycle Manager uses the production object as the reference object and set the CREATE parameter code to R. |
| T         | Use the test object as a reference for parameters when creating this object. If the object does not exist in test, use the named reference object or the operating system defaults.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| S         | Use the command in the TURNOVER® Lifecycle Manager Object Type file for the create object command.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| R         | Use the named object (entered in the space provided on the panel) as a reference when creating this object. If the reference object does not exist, TURNOVER® Lifecycle Manager issues an error message.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| C         | Parameters will be specified explicitly when a line is added to form. You can set up reference objects in your production library (such as REFPPF, REFLF, REFMSGF, REFPGM, REFCMD, REFDSPPF, REFPRTF, and so on), and then type REF* in the compile parameter reference field. TURNOVER® Lifecycle Manager parses together REF and the object type code (for example, "CLP") and then look for that reference object. TURNOVER® Lifecycle Manager uses that as the reference object.                                                                                                                                                                                                                                                                                          |

---

## Initial object authority

See *'Security recommendations' on page 149*

---

## Logical files

You do not have to explicitly delete a logical file over a physical file before recreating or deleting the physical file. TURNOVER® Lifecycle Manager deletes logical files automatically. When you enter a physical file on a TURNOVER® Lifecycle Manager form, TURNOVER® Lifecycle Manager inserts a line on the form for each logical over the physical being deleted or replaced. This is done

after you have entered all the lines on the TURNOVER® Lifecycle Manager form. When you exit from the Form Maintenance panel, a message appears: "Checking data base relations for PF on form". TURNOVER® Lifecycle Manager verifies and adds entries for logical files at this time. Logical files already on the form are not duplicated.

## How the logical file logic works

Every time a replace or delete physical file line is entered, TURNOVER® Lifecycle Manager checks for logical files over the existing physical file.

### Does a logical file exist?

|      |                                                                                               |
|------|-----------------------------------------------------------------------------------------------|
| Yes: | Is there a delete for this logical file on this form?                                         |
| Yes: | Do nothing.                                                                                   |
| No:  | Is the logical file created earlier on this form?                                             |
| Yes: | Move the line to create the logical file to the line following the create physical file line. |
| No:  | TURNOVER® Lifecycle Manager inserts a line to create the logical file.                        |

Only files that you have changed need to be explicitly referenced on the form. All dependent logical files, even those not being changed, will be added to the form automatically. This ensures that the creation of every object in production can be associated with a TURNOVER® Lifecycle Manager form.

## Deleting a logical file

When you enter a line to delete a logical file on the TURNOVER® Lifecycle Manager form, TURNOVER® Lifecycle Manager assumes that you will take care of recreating it yourself. In this case, TURNOVER® Lifecycle Manager does not generate a new line to recreate the logical file automatically.

## Deleting a physical file with logicals

After you finish entering a TURNOVER® Lifecycle Manager form, TURNOVER® Lifecycle Manager inserts logical file delete lines for any physical file with logicals over it that is being deleted.

## Multiple-membered physical files

TURNOVER® Lifecycle Manager preserves all members of an existing physical file. You are responsible for adding new members or removing obsolete members. You can do so by putting a post-run command on the form.

## Existing physical files

If a multiple-membered file is being replaced, TURNOVER® Lifecycle Manager recreates the file with all the members that were in the physical file. All of the members are preserved and copied in the new version of the file using the copy command with the format options `*MAP` and `*DROP`.

If you need to add or remove members in the file, use the `ADDPFM`, `ADDLFM`, or `RMVM` command. This can be done on the same TURNOVER® Lifecycle Manager form by entering pre-run or post-run commands.

## New physical files

If the file is new, TURNOVER® Lifecycle Manager creates the physical file with one member (member name equal to file name). You must add additional members to the file after the file is created using the `ADDPFM` command. TURNOVER® Lifecycle Manager does not automatically replicate members from the test file. The `ADDPFM` command can be entered as a post-run command of the same TURNOVER® Lifecycle Manager form.

If data needs to be copied into a member, you can call a program, use a copy command or use the `CPYTSTDTA` command. For more information, see *'Commands' on page 409*.

---

## Multiple-membered logical files

TURNOVER® Lifecycle Manager preserves all members of an existing logical file. You are responsible for adding or removing members.

## Existing logical files

If the multiple-membered logical file is being replaced, TURNOVER® Lifecycle Manager recreates the file with all the members that were in the original logical file.

If you need to add or remove members in the file, use the `ADDLFM` or `RMVM` command. This can be done on the same TURNOVER® Lifecycle Manager form by entering pre-run or post-run commands.

---

## New logical files

If the logical file is new, TURNOVER® Lifecycle Manager adds one member (member name is file name) to the logical after the logical file is created. Additional members must be added to the logical file by prompting the line on the form or, after the file is created, using the `ADDLFM` command. TURNOVER® Lifecycle Manager does not automatically replicate members from the test file. The `ADDLFM` command can be entered as a post-run command on the same form.

---

## Duplicating and moving logicals

Logical files are created from source using the `CRTLF` command as if you had specified `CSCO` to ensure proper creation over the correct physical files. However, if you specify a method of `MO`, `CSMO`, `CD`, or `CSCD` in your application definition type code entry, this will override the rule of recreating logicals from source.

Exercise caution when overriding the recreate rule.

To also compile logical files on production computers, TURNOVER® Lifecycle Manager always distributes LF source with the files. On the target system, the application definition field, **Target Source library**, must contain the name of a valid target source library, or it must contain the value `QTEMP` (no blanks are allowed). TURNOVER® Lifecycle Manager then copies the source to the specified target source library. Unless the type code specifies a different method, TURNOVER® Lifecycle Manager creates the logical file from the source.

---

## Journaling files

A file that is being journaled can be deleted and recreated. When the file is deleted, an "MD" entry is written to the journal receiver by the operating system to indicate that the "member was deleted".

TURNOVER® Lifecycle Manager automatically restarts journaling to any journaled file or access path that is being replaced. TURNOVER® Lifecycle Manager restarts journaling after the journaled file is created and repopulated with data or, in the case of a logical file, after the access path is rebuilt.

TURNOVER® Lifecycle Manager ends journaling before moving a replaced file to the archive library if it does not find the 7-character data area, `TENDJRN`, in the TURNOVER® Lifecycle Manager library list. TURNOVER® Lifecycle Manager does not end journaling if it finds the `TENDJRN` data area in the TURNOVER® Lifecycle Manager library list and if the data area contains a value other than `*ENDJRN`. This behavior also applies to the archiving of journaled files in explode libraries.

You are responsible for starting journaling for any file or access path being added. You can do this with a post-run command. If you are journaling files in a series of explode libraries, use the `"&ALLTGT"` (including the quotes) parameter. For more information, see *'Pre-run and post-run commands' on page 66*.

If you are using a journal for backup/recovery, save the file or its library *before* and *after* the file is recreated, to establish the necessary checkpoints entries in the journal receiver.

---

## Promoting programs

Programs are created, moved or duplicated as described in Source Compiled Objects diagram. (See earlier pages.)

If you specify `REPLACE (*YES)` in the create parameter, TURNOVER® Lifecycle Manager duplicates the program into the archive library before it recreates the object. This ensures that use of the program is not interrupted during the compile and replace process. (Users do not have to sign off so you can change a program.) You must explicitly code the command in TURNOVER® Lifecycle Manager's type file with `REPLACE (*YES)` and the actual owner of the program must submit the form for these features to work correctly.

---

## IBM i menus

IBM i menu object types are supported by TURNOVER® Lifecycle Manager. During checkout, TURNOVER® Lifecycle Manager determines the type of menu (message file or program) and copies the message file and other menu components or source to the development library. When you build the TURNOVER® Lifecycle Manager form using the reference-build feature, TURNOVER® Lifecycle Manager includes lines for each menu component on the form. For more information, see the *'Menus, DFUs, and queries' on page 360*.

---

## S/36 menus and DFUs

TURNOVER® Lifecycle Manager provides a special checkout option for S/36 menus and DFUs. When you create a form using the reference build feature, TURNOVER® Lifecycle Manager includes lines for each menu or DFU component on the form. For more information, see the *'Menus, DFUs, and queries' on page 360*.

---

## Processing commands

When promoting a command (`*CMD`), TURNOVER® Lifecycle Manager retrieves the values of the control processing program (`CPP`) and validity checking program (`VLDCKR`) from the `*CMD` object in the *from* library. If the `*CMD` does not exist in the *from* library, TURNOVER® Lifecycle Manager retrieves these values from the object that was just replaced and is now in the archive (`TnnnnnnnD`) library. If neither object exists, TURNOVER® Lifecycle Manager defaults the `CPP` parameter to the name of the command and the `VLDCKR` parameter to `*NONE`.

Therefore, if you are using method `CSCO` (Copy Source, Compile Object), it does not matter what values are in the `CPP` and `VLDCKR` parameters on the `CRTCMD` (Create command) command itself. Furthermore, the **Create Parm**s value (`P`, `T`, `R`, `S`, or `C`) for the form line has nothing to do with the command processing.

The way you can control the `CPP` and `VLDCKR` parameters is to correctly create the `*CMD` in the library from which you are promoting it.

---

## Pre-run and post-run commands

Pre-run commands run at the beginning of a TURNOVER® Lifecycle Manager form job (`*PREFORM`) or just before an object on the form, associated with the command, is promoted (`*PRELINE`).

Post-run commands run after all objects are promoted (POSTFORM), or just after an object associated with the command is promoted (POSTLINE).

Exactly when the command is run depends on the parameter you set when you enter the pre/post-run command on the TURNOVER® Lifecycle Manager form.

No interactive or program-only commands (commands such as RTVSYSVAL) are permissible. Commands must be able to be run in batch using the QCMDEXC or QCAEXEC programs. Override commands, such as OVRDBF, are permissible.

Commands that would interfere with the TURNOVER® Lifecycle Manager run are not allowed. For example, RNMOBJ of an object that is being replaced in a TURNOVER® Lifecycle Manager before a TURNOVER® Lifecycle Manager starts would cause an error. RNMOBJ after a TURNOVER® Lifecycle Manager runs is acceptable technically, but we suggest deleting and adding the object with a new name on the form so that the object history reflects this change correctly.

Both pre-run and post-run commands have a **Run on system** parameter for specifying on which systems the command is to be run. The values for this parameter are \*ALL (the default), \*LOCAL, \*REMOTE, or a system name.

## Pre-run commands

Pre-run commands can be conditioned such that the TURNOVER® Lifecycle Manager can be either terminated or continued, depending on the successful execution of the command. These commands can run using the library list under which the job is running (\*CURRENT) or the library list specified for the applications (\*APPL). Which library list is used depends on parameter you set when you enter the pre-run command on the TURNOVER® Lifecycle Manager form.

## Post-run commands

Post-run commands can be conditioned such that subsequent commands can be executed or not, depending on the successful execution of the command.

A post-run command failure appears as a warning message in the TURNOVER® Lifecycle Manager Log Report, but does not cause a rollback of the TURNOVER® Lifecycle Manager or any part of it. The TURNOVER® Lifecycle Manager is considered complete by the time the post-run commands are run.

Post-run commands can run using the library list under which the job is running (\*CURRENT) or the library list specified for the applications (\*APPL). Which library list is used depends on parameter you set when you enter the post-run command on the TURNOVER® Lifecycle Manager form.

---

## Pre-run and post-run command scenarios

You might need to set up TURNOVER® Lifecycle Manager to run a special conversion program after a physical file is replaced. (TURNOVER® Lifecycle Manager copies an old file to a new file using `FMTOPT`, with `*MAP`, `*DROP`, or `*NOCHK`, if you want it to.) In this case, enter a `CRTDUPOBJ` command to be executed before the first TURNOVER® Lifecycle Manager step and a `CALL` to a conversion program after the last TURNOVER® Lifecycle Manager step.

If you have a big file, you might want to move it to another library and duplicate it back into production using pre-run commands, and then copy it into the new version file using a post-run command (`CPYF` with `*MAP *DROP`) or a conversion program.

You might want to do a `SAVLIB` or `SAVOBJ` before starting a TURNOVER® Lifecycle Manager. This can be entered as a pre-run command.

You might have a program that uses a temporary file. You can create this in `QTEMP` with a pre-run command.

You might want to copy data from a test file to production. This can be done using the `CPYTSTDTA` command as a post-run command. For more information, see ['Commands' on page 409](#).

If you use the Explode data library feature, any post-run command that contains `"&ALLTGTT"` (including the quotes) in a library parameter will be applied repetitively to the primary target library and to each explode library.

If you have authority, you might want to grant additional authorities. This can be done either at the beginning or end of the promotion.

For information about including pre- and post-run commands on a TURNOVER® Lifecycle Manager form, see "Adding pre-run and post-run commands" in the *TURNOVER® Lifecycle Manager User Guide*.

---

## Deleting objects

To delete an object from production, add the item to the TURNOVER® Lifecycle Manager form set the Action code to `D` (Delete).

If you are promoting the object through more than one application level and the object does not exist in any Target library at any level, TURNOVER® Lifecycle Manager will ignore the TURNOVER® Lifecycle Manager form line.

TURNOVER® Lifecycle Manager moves the object to the Archive library and archives any existing source, before it deletes the object and source from the Target library.

---

## Form copy and object delete rules

This section defines the rules TURNOVER® Lifecycle Manager follows when:



- copying forms from one application level to a higher application level
- deciding whether or not to delete test From objects when the TURNOVER® Lifecycle Manager form is run.

## Background

TURNOVER® Lifecycle Manager adds lines to forms automatically during a cross-reference check (if you use a cross-reference facility) or by checking database relationships for physical files. TURNOVER® Lifecycle Manager adds a line to the form for each related object it finds, and then sets the method for these lines to `COPR`, which compiles the object into the target library for this level of the application, using the source available either in the target source library *or at a higher level*.

For example, if you are promoting a file to an intermediate test level, TURNOVER® Lifecycle Manager looks for programs created over that file at that test level and at the higher production level. If a related object is found, it is added to the form as method `COPR` to be created in the test library from the source in the production library.

## Form copy rule

When you copy a form from one level or application to another, and that form contains objects that were created at the current level using method `COPR`, then the following rule applies:

### Copy rule

If an object on the form you are copying was added to the form with method `COPR`, *and* if the object is not checked out to the new form's programmer under the same project task reference as the form being copied, the object line is dropped from the new form. If that same object is present in the cross-reference file for the target level, the object is again added with method `COPR` during the cross-reference check that occurs when you exit from the new copy of the form.

This is necessary because an object that exists at one level may not exist at the next higher level and, therefore, should not be recreated there. Also, a logical file might be built over libraries outside of the application. These will not convert correctly during the form copy; therefore the logical file form lines must be deleted and re-added after the copy completes.

TURNOVER® Lifecycle Manager issues this message when it drops an object during the copy process:

```
Message . . . .: TUR0963: Object objectname on form formnumber was not
copied to new form. Reason code=n.
```

Interactively, you can see the reason code information by positioning your cursor on the message and pressing F1. If you submitted your copy to batch, you should see the reason codes in the joblog.

| Code | Description                                                                                                                                      |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | The object is a logical file that was recompiled (method <code>COPR</code> ) in a library other than the application data library.               |
| 2    | The object being recompiled (method <code>COPR</code> ) is not checked out to the programmer and reference that are on the form you are copying. |
| 3    | The object did not promote successfully on a form that you are copying to the next level of the application.                                     |
| 4    | The object was promoted successfully on a form that ran with errors, and it is being copied to the same level of the application.                |
| 5    | A matching type code was not found.                                                                                                              |

## Delete test object rule

During a form run, TURNOVER® Lifecycle Manager applies the following rules when deciding whether to delete the test object in the From library:

If **Delete test object** is Y, then...

If Global parameter **Use delete test object rule** is N

No further action required.

Else

If the **From object** is checked out to another programmer or the **From object** is checked out under a different project reference

Set **Delete test object** to N.

Else

If **From object type** is a program and a cross-reference exists for the From-application level then, for each file used by the program, do the following:

If the file exists in the from-library

*and* the same file name exists in the target library at a higher-level library

*and* the level-id of the file in the from-library is not equal to the level-id of the file at a higher level

Set **Delete test object** to N.

If source is promoted and **Delete test source** is Y, the from-source is deleted.

**Note** If the program is not deleted because the dependent file was different in the **From library** than in the **Target** or higher-level *library*, eventually the program *will* be deleted when the changed file is promoted and the dependent programs are recreated in the target library.

If during the form run TURNOVER® Lifecycle Manager found it necessary to change **Delete test object** from Y to N, this message appears in the joblog:

```
Message: . . . .: TUR0965: Delete From Object changed to N for  
objectname. Reason code n.
```

### Reason codes

| Code | Description                                                                                                                                    |
|------|------------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | The object is currently checked out to another programmer, or to the same programmer under another project reference.                          |
| 2    | Deleting the <b>From object</b> might cause level checks in the <b>From</b> environment. Check the file references of the <b>From object</b> . |

# 5 Source Stream Files

---

## Introduction to source stream files

Creating and editing source files from IBM i's integrated file system (IFS) can offer organizations greater flexibility and a more modern approach to development. For instance, the use of open source tooling may be introduced to extend workflow capabilities that would otherwise be unattainable with source code residing in the native (QSYS) file system.

With IBM i v6.1, ILE compiler copy/include mechanisms for RPG, Cobol and C can support both source members (QSYS) and stream files (IFS). /COPY and /INCLUDE can include sources from both file systems, regardless of where the parent source is maintained.

TURNOVER® Lifecycle Manager v2.1 introduces comprehensive support for source stream files across the software development lifecycle, allowing users to utilize IFS-based source code in the same, familiar way, as native source code.

---

## Pre-requisites

The following pre-requisites for using source stream files in TURNOVER® Lifecycle Manager exist:

- IBM i 7.1, or later
- TURNOVER® Lifecycle Manager v2.1, or later

---

## Supported file systems

IBM i supports several types of file system within the integrated file system. Not all of them are supported by TURNOVER to hold source stream files and some have considerations with regard to mixed-case naming conventions. If you attempt to use a file system that is not supported, results are undetermined and technical support may not be provided.

| File system | Description                                | Supported by<br>TURNOVER | Case-sensitive |
|-------------|--------------------------------------------|--------------------------|----------------|
| /QSYS.LIB   | Native objects, source and database tables | Yes                      | No             |

---

| File system   | Description                                       | Supported by<br>TURNOVER | Case-sensitive   |
|---------------|---------------------------------------------------|--------------------------|------------------|
| / (root)      | Hierarchical stream files similar to Windows      | Yes <sup>4</sup>         | No               |
| /QOpenSys     | Hierarchical stream files similar to *nix (POSIX) | Yes                      | Yes <sup>1</sup> |
| /QNTC         | Remote Windows (NT) servers                       | No <sup>2</sup>          | No               |
| /QFileSvr.400 | Remote IBM i file systems                         | No <sup>2</sup>          | Mixed            |
| /QDLS         | OfficeVision/400 document library system          | No                       | No               |
| /QOpt         | Mounted optical media                             | No                       | No               |
| <UDFS>        | User defined file systems                         | Yes <sup>3</sup>         | See <sup>1</sup> |

**Note 1:** If you store source stream files in the /QOpenSys file system or in a user defined file system that you have chosen to respect case-sensitive names then you must be sure to add any worklist item object names using the correct case of the source file since this will be used to locate the source for the object. If the case of the object name is not the same as the source file then the source will not be found. For non-case-sensitive file systems the case of the name is not important since the object name and source file name will match regardless of case.

In case-sensitive file systems it may be possible to have files with the same intrinsic name in mixed case. For example source files Pgm1.RPGLE and pgm1.rpgle will both create the native ILE RPG program object PGM1.

TURNOVER adheres to a strict principle of one object to one source file with the same name as the object being created and the same source attribute suffix that would be used if the source were stored in a native source physical file.

A global type code that uses the \*SRCSTMF change scheme determines the case of the source file suffix and must be used consistently. For example, if the type code is intended to be used with source stored in a case-sensitive file system then the source attribute should be entered in the case to be used for names in that file system. Whatever case you enter the source attribute in the global type code is the expected case of the source file suffix in the file system. For example, if the global type code specifies the source attribute as 'RPGLE' then the source file suffix must also be objectName.RPGLE. If the source file name is objectName.rpgle then it will not be found by TURNOVER.

Because of these complications it is highly recommended that you avoid using a case-sensitive file system to store source stream files if you can. You may need to use a case-sensitive file system if you have ported code from another system such as C source code that refers to include files that have case-sensitive names and that you cannot port into the root file system for some reason. Some open source software requires source code to be store in POSIX style file systems for instance.

If you can use a non-case-sensitive file system your life will be much easier.

**Note 2:** The /QFileSvr.400 file system provides a method of accessing the integrated file systems of remote IBM i servers.

The /QNTC file system provides a method of accessing the file system of remote Windows (NT) servers.

While you may technically be able to work with source folders on a remote server through these file systems we do not recommend that you do so as they are officially unsupported by TURNOVER.

**Note 3:** If you store source in a user-defined file system (UDFS) then you must set a fixed base path in each level of the application definition where you intend for the UDFS to be mounted. The source folder name for each type code in the application level must then reflect the path within the UDFS to store the source files.

You are responsible for making sure that the UDFS is mounted correctly over the designated base path before using any TURNOVER function that may manipulate the content of folders, such as creating missing items, checking source out, compiling objects and running forms. If the UDFS is not mounted or is not mounted on the designated base path then TURNOVER will be unable to locate the source files.

Mounting a UDFS over a base path masks the original content of the base path replacing it with the file structure of the UDFS. If you operate TURNOVER functions that create missing items, for example without mounting the file system first, then the missing item folders will be created in the base file system folder and not the UDFS. When you mount the UDFS the newly created folders will be masked and TURNOVER will not find them.

**Note 4:** It is highly recommended that source stream files are stored in folders in the root (/) file system where case-sensitive file names are not an issue. You are free to name your source folders and files in any case you choose. Source file names PGM1.RPGLE or pgm1.rpgle or Pgm1.Rpg are all valid and will create the ILE RPG program object PGM1.

---

## Recommended interfaces

Use of the TURNOVER® Client and/or TURNOVER® Lifecycle Manager's Eclipse plugins is suggested for all source stream file setup within TURNOVER® Lifecycle Manager. This will not only provide a much improved user experience but will also ensure a more expedient set up process.

**Note** Please refer to the *TURNOVER® Lifecycle Manager Getting Started* for installation and upgrade instructions.

---

## Source naming convention

The implementation of source stream file names is designed to reflect the native object name and type that the source creates.

The source file suffix will reflect the source member attribute (CLLE, RPGLE, CBLLE etc.) that, in a native environment, is used to store the source as defined by the associated TURNOVER® Lifecycle Manager type code.

For example, the source stream file for ILE RPG program *MYPROGAM* would be:

```
/path/to/application/source/myprogram.rpgle
```

---

## Source path convention

There are several areas within TURNOVER® Lifecycle Manager in which you can specify a stream file source path. This includes programmer properties, application definition levels and project definitions. In each case, it is important that when specifying a fixed path, the path must start with a forward slash "/" but never end with a forward slash. As an example, for programmer Joe Bloggs (profile: BLOGGS), the Source path value should be similar to that below:

```
/HOME/PGMR/BLOGGS
```

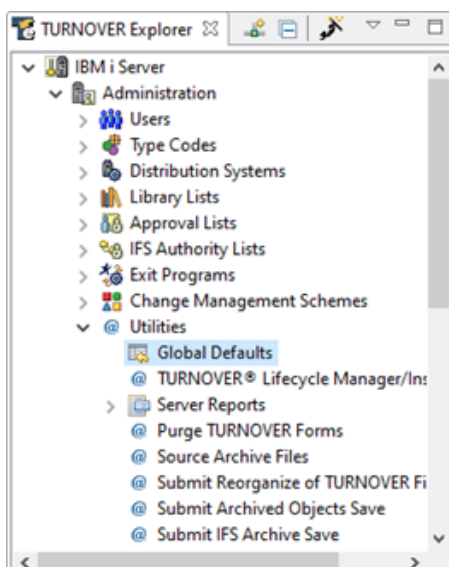
In this example "/HOME" is a top-level folder to act as a container for all other *Source path* references to provide a degree of structure and organization to the IFS. You can choose your own naming convention as you see fit, however, you should not attempt to use "/TURNOVER" as this is reserved for a set of specific actions/requirements within TURNOVER® Lifecycle Manager.

---

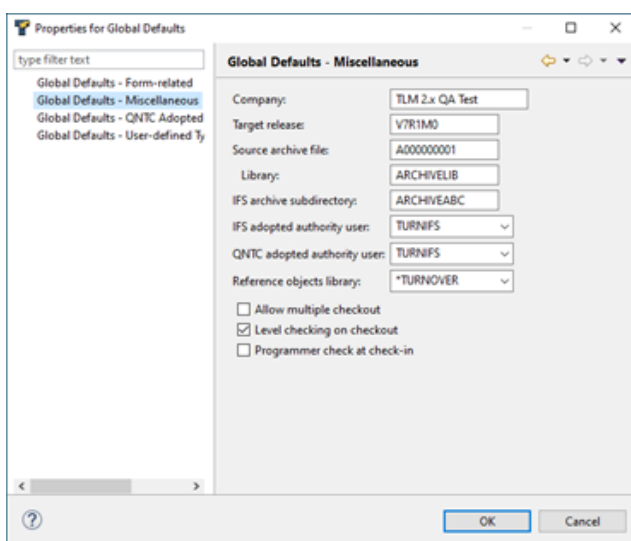
## Source stream file management configuration and IFS folder structure

TURNOVER® Lifecycle Manager will use the existing *source archive file* and *source archive library* values, as specified under the Miscellaneous Global Defaults, for source stream file requirements. You should familiarize yourself with these values from the outset.

Using TURNOVER® Lifecycle Manager's Eclipse plugins or the TURNOVER® Client, under the TURNOVER Explorer: Administration > Utilities > Global Defaults



**Right-click** on the Global Defaults entry in the TURNOVER Explorer and select **Browse/Edit Global Defaults**. At the resulting dialog, select the **Global Defaults – Miscellaneous** entry from the list of options on the left. You will be presented with a set of fields as illustrated in the image below:



In order to avoid confusion with the IFS (Integrated File System) setup for non-native artifacts, the existing values for the *Source archive file* and *Library* will be used for QSYS-managed source and IFS stream source implementations as well.

In the case of the IFS, the resulting folder structure will be created under /TURNOVER/ARCHIVE/ when first needed. For instance, using the values as illustrated above, this will create an IFS folder structure as follows:

```
/TURNOVER/ARCHIVE/ARCHIVELIB/A000000001/
```

In addition, TURNOVER® Lifecycle Manager will make use of the following pre-defined IFS folder structures for specific stream source file actions that it performs, such as:

```
/TURNOVER/DISTRIBUTION (for form distribution to another system)
```

```
/TURNOVER/FORMARC (for form promotion archiving)
```



/TURNOVER/FORMWORK (for transient stream files created when running forms)

/TURNOVER/RECEIVE (for receiving form distributions i.e. on a target system)

## Source stream file type codes

TURNOVER® Lifecycle Manager provides a number of pre-configured type codes for commonly used source stream file types. These will be installed when performing a new installation of TURNOVER® Lifecycle Manager.

When upgrading from a previous version of TURNOVER®, each source stream file-related type code will be installed as long as there is not already a type code within your installation that uses the same name. In the event of a type code name already being used, the upgrade routine will simply leave the existing type code in place and will move on to the next source stream file type code. For any source stream file type codes that could not be created at the time of the upgrade, you can copy them from the installation library at a later time, assigning an alternative type code name as required.

TURNOVER enables OPM programs and other native object types such as display files, print files, physical and logical files that would normally require source to be stored in native source physical files to be created from source stream files.

TURNOVER achieves this by automatically copying the source stream file content to a temporary native source member in the QTEMP library immediately before compiling the object and then discarding the temporary copy.

In order to do this, type codes that create native objects that require a native source file member should define the create command SRCFILE parameter as SRCFILE(QTEMP/"&QF"). The new "&QF" substitution variable will be replaced by the global type code source file name at compile time.

Source stream file type codes that are provided are listed below:

| Type Code Name | Description                     |
|----------------|---------------------------------|
| CBLS           | OPM COBOL programs              |
| CBLLS          | ILE COBOL program               |
| CBLMOS         | ILE COBOL Module                |
| CBLSQS         | ILE COBOL SQL program           |
| CLES           | ILE C program                   |
| CLEMOS         | ILE C module                    |
| CLLENS         | ILE CL program prior to V7R4M0  |
| CLLES          | ILE CL program V7R4M0 and above |

| Type Code Name | Description                      |
|----------------|----------------------------------|
| CLPS           | OPM CL program                   |
| CLPMNS         | ILE CL module prior to V7R4M0    |
| CLPMOS         | ILE CL module V7R4M0 and above   |
| CMDNS          | Command prior to V7R4M0          |
| CMDS           | Command V7R4M0 and above         |
| DSPFS          | Display file                     |
| ICFFS          | Intra-system communications file |
| LFS            | Logical file                     |
| MNDSPS         | DDS menu                         |
| MNUIMS         | UIM menu                         |
| PFS            | Physical file                    |
| PLIS           | OPM PL/1 program                 |
| PNLGRS         | Panel group                      |
| PRTFS          | Print file                       |
| RPGS           | OPM RPG program                  |
| RPGLES         | ILE RPG program                  |
| RPGMOS         | ILE RPG module                   |
| RPGSQS         | OPM RPG SQL program              |
| SQSCBI         | ILE COBOL SQL program            |
| SQSCBM         | ILE COBOL SQL module             |
| SQSCBS         | ILE COBOL SQL service program    |
| SQSC           | ILE SQL C program                |
| SQSCM          | ILE SQL C module                 |
| SQSCS          | ILE SQL C service program        |
| SQSRPI         | ILE RPG SQL program              |

| Type Code Name | Description                          |
|----------------|--------------------------------------|
| SQSRPM         | ILE RPG SQL module                   |
| SQSRPS         | ILE RPG SQL service program          |
| ILESRS         | ILE service program V7R4M0 and above |
| ILSSRS         | ILE service program prior to V7R4M0  |

As a general rule, the type code naming convention for source stream types is based upon TURNOVER® Lifecycle Manager's native programming language equivalent with the addition of a suffix "S".

As an example, the default *ILE RPG Program* type code that ships with TURNOVER® Lifecycle Manager is **RPGLE**. Therefore, the source stream file equivalent is **RPGLES**.

There is an exception, however, in the case of SQL-related type codes. For these type codes, rather than provide a suffix of "S" to denote source stream file, the "S" takes place of the "L" in "SQL". In other words, for the source stream file equivalent to the *ILE RPG SQL Program* type code (**SQLRPI**), the type code takes the name **SQSRPI**.

One final exception concerns type codes that require a pre-v7.4 creation command as well as a v7.4 equivalent. An example of this can be found with the stream source equivalent of the *Command* type code. In this case, the native **CMD** type code exists and, in addition, **CMDNS** for pre-v7.4 and **CMDS** for v7.4. Where possible, a consistent convention has been applied but there are a handful of instances where this has not been possible.

An example of a pre-configured TURNOVER® Lifecycle Manager source stream file type code is provided in the image below:

In this example, the stream file type code is attributed a *Scheme* value of **\*SRCSTMF** – this is the case with all source stream file-related type codes.

Additionally, the source stream file type codes come preconfigured with an appropriate default creation command. Within the creation command there are a number of substitution variables that can be used. These are:

**&QF** (Global type code source file name. This will be used by the compile process to copy the source stream file into a QTEMP native source physical file with the same name as the global type code)

**&AT** (Source file attribute)

**&FP** (From path)

**&SP** (Source file path)

There are two ways of defining the stream file path within a TURNOVER® Lifecycle Manager type code. You can either use the **&SP** variable which will write out the entire source stream file path:

/library/file/member.attribute

or, alternatively, you can use a collection of individual variables as follows:

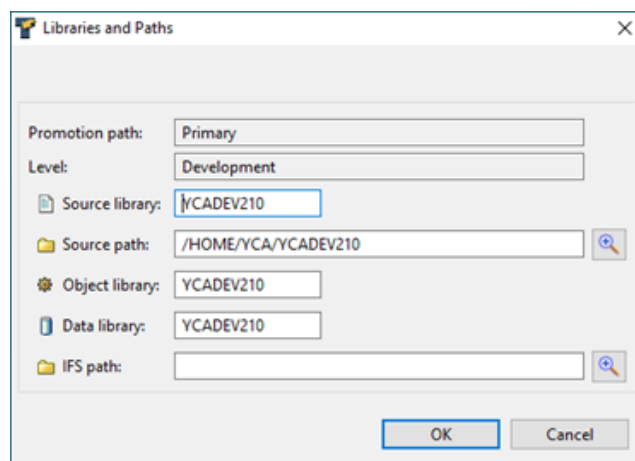
**"&SL"/"&SF"/"&SM".&AT.**

In the case of the example directly above, **"&SL"** is defined by the application level; **"&SF"** is defined by the type code; and **"&SM"** and **"&AT"** are defined by the object type.

## Application Definitions

UNICOM recommends using the TURNOVER® Client/TURNOVER® Lifecycle Manager Eclipse plugins for creating new application definitions that involve source stream files, as well as for modifying existing application definitions that are being updated to manage source stream files.

TURNOVER® Lifecycle Manager now allows you to specify a base *Source path* value for each level that is defined within new and existing application definitions:



Valid entries for the *Source path* include **\*PGMR**, **\*PROJECT** and path construct as indicated above. A couple of important points to note regarding path constructs are that:

- /TURNOVER is a reserved folder location and should therefore not be used as a top level folder location for source path
- The source path value must **not** end with a forward slash
- You must enter a source path even if you are not using stream file type codes. The root (/) folder will be assumed as the default if no explicit path is provided.

In the case of **\*PGMR** and **\*PROJECT**, the source path will be derived from the Programmer or Project attributes, respectively.

Base *Source paths* should not be duplicated across your application definitions. There are times when TURNOVER® Lifecycle Manager has to identify the correct application from a source path. If the same source path is used by more than one application, for example sharing a common QA path, the correct application will only be identified by differentiating the source path, for example if the application code is included in the path construct. The exception is when defining an emergency application, which has to promote into the same top-level source path as its standard application counterpart.

The *Source path* must be pre-created using the application *Create Missing Items* function before they can be used in the same manner as libraries are created. If you attempt to check-out an object source without creating these locations first you will receive an error.

---

## Level properties

An appropriate authority list should be specified for both the *Target authority list* and *Development authority list* options under the Line defaults section of the application definition *Level properties* panel:

The screenshot shows the 'Properties for 10: dev - test' dialog box with the 'Line defaults' tab selected. The left sidebar lists various configuration categories, with 'Line defaults' highlighted. The main area is divided into two sections: 'IBM i' and 'IFS'. Each section contains several configuration items, each with a 'Locked' checkbox.

| Section | Property                  | Value                               | Locked                   |
|---------|---------------------------|-------------------------------------|--------------------------|
| IBM i   | Promotion method          | Copy source, compile object (CSCO)  | <input type="checkbox"/> |
|         | Update object description | <input type="checkbox"/>            | <input type="checkbox"/> |
|         | Check for level check     | <input type="checkbox"/>            | <input type="checkbox"/> |
|         | Create parameter          | Use Target Object                   | <input type="checkbox"/> |
|         | Reference object          | *REF                                | <input type="checkbox"/> |
|         | Library                   | XINFREF                             | <input type="checkbox"/> |
|         | Authority parameter       | Use Target Object                   | <input type="checkbox"/> |
|         | Reference object          | *REF                                | <input type="checkbox"/> |
|         | Library                   | XINFREF                             | <input type="checkbox"/> |
|         | Delete from object        | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| IFS     | Delete from source        | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
|         | Programmer                | *CHECKOUT                           | <input type="checkbox"/> |
|         | Project task reference    |                                     | <input type="checkbox"/> |
|         | Promotion method          | CPY                                 | <input type="checkbox"/> |

Additional IFS properties shown below the table:

| Property                   | Value                    | Locked                              |
|----------------------------|--------------------------|-------------------------------------|
| Delete from object         | <input type="checkbox"/> | <input type="checkbox"/>            |
| Target authority list      | YOURGRP                  | <input checked="" type="checkbox"/> |
| Development authority list | YOURGRP                  | <input type="checkbox"/>            |

The specified authority list will then be used by the *TCRTSSF* command (parameter *AUTL*) when running *Create Missing Items* (at a later stage) to control ownership of the paths and therefore prevent unauthorized access to the IFS source locations that relate to the application level.

---

## Type code defaults

The *Source file* and *Development source* file fields are now extended in length for when application level type codes relate to source stream files (scheme \*SRCSTMF) to accommodate potentially longer folder names:

RPGLES Type Code Defaults

Level: Production(10)

Type code: RPGLES (\*PGM - Stream ILE RPG program)

Source file: QRPGLESRC ☐ Locked

Development source file: QRPGLESRC ☐ Locked

These properties can be specified uniquely for this type code, or inherited from the application level properties. To specify a property uniquely for this type code, toggle the button to the right.

< Delete from source: ☒ ☒ Locked

< Delete from object: ☒ ☐ Locked

< Promotion method: Copy source, compile object (CSCO) ☒ Locked

Object Attributes

< Create parameter: Use Target Object ☐ Locked

< Reference object:  ☐ Locked

< Library:  ☐ Locked

Object Authority

< Authority parameter: Use Target Object ☐ Locked

< Reference object:  ☐ Locked

< Library:  ☐ Locked

Distribute: Use application/system defaults

OK Cancel

This allows for a hierarchical structure or longer source folder names to be specified, where required, which will then be appended to the base path for the promotion level e.g.:

*/basepath/QRPGLESRC/X/Y*

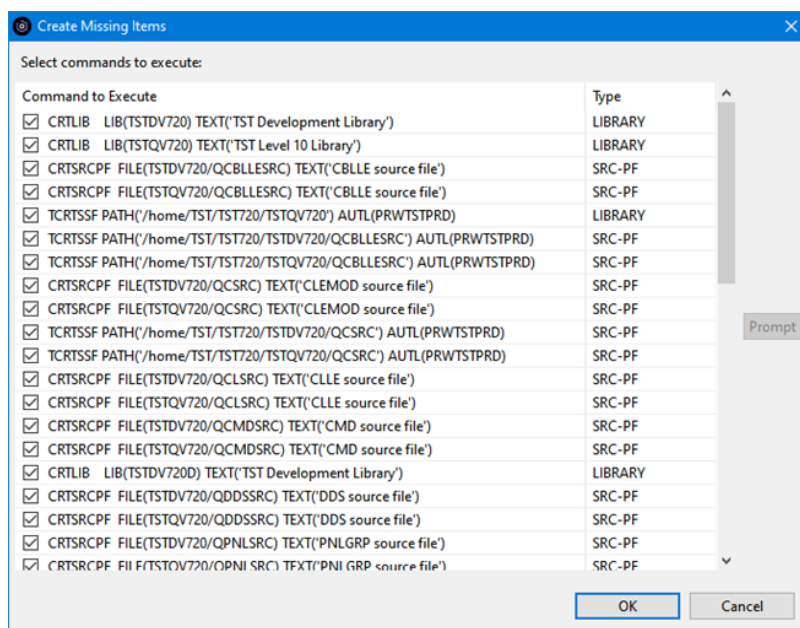
*/basepath/my\_longer\_source\_folder\_name\_goes\_here*

---

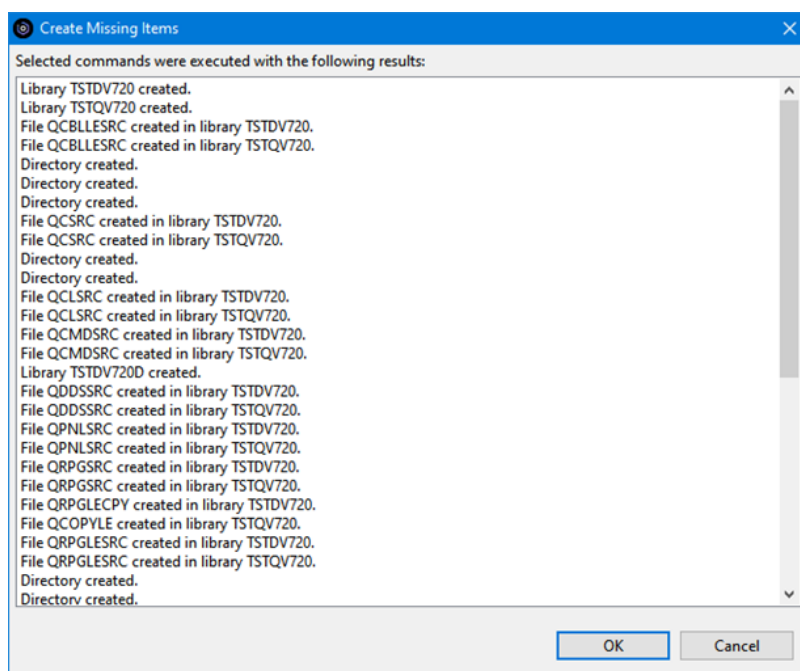
## Create missing items

When you have finished setting up your application definition, remember to save and check the application definition back in.

You can then **right-click** on the application and select the *Create Missing Items* option from the menu. You will be presented with a screen along the following lines:



Press **OK** to continue. You will receive confirmation of the items that have been created:



## Submit initial build

After completing the application definition setup and running the *Create Missing Items* process, the *Submit Initial Build* process needs to be run against each application level. In the context of source stream files, this will identify existing libraries and objects that were created by source stream files prior to TURNOVER's involvement, and will extract the source file information and record it as part of the application definition base audit report.

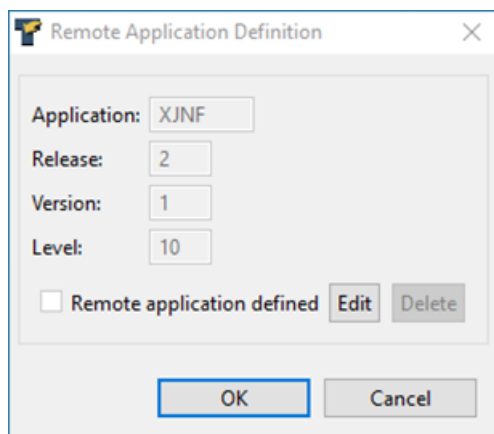
If any of the libraries contain objects that were previously built from source stream files this information will also be collected in the base audit report.

---

## Remote systems

If any of your application levels make use of a remote system, the remote application setup will default to the same setup as its corresponding base application level, however, this can be overridden if required.

To override the remote application definition settings, check out the application definition and then **right-click** on the required application level and select *Remote Application Definition* from the corresponding popup menu. You will be presented with a dialog similar to that below:

A screenshot of the 'Remote Application Definition' dialog box. The dialog has a title bar with a close button. Inside, there are four input fields: 'Application:' with the value 'XJNF', 'Release:' with the value '2', 'Version:' with the value '1', and 'Level:' with the value '10'. Below these fields is a checkbox labeled 'Remote application defined' which is currently unchecked. To the right of the checkbox are 'Edit' and 'Delete' buttons. At the bottom of the dialog are 'OK' and 'Cancel' buttons.

Click the **Edit** button.

You may be presented with a warning message informing you that the remote application definition does not exist for the application and asking whether you would like the remote application definition to be created. Click the **Yes** button.

You will be presented with the application properties panel. Review and make changes as necessary. You may want to specify a different *Target source path* for source stream files under the *Library and path defaults* section. Similarly, you may want to change the *IFS Target authority list* under the Line defaults section.

When all changes are complete, click the **OK** button.

Remember to check the application definition back in when you are ready.

---

## Project administration

TURNOVER now allows you to specify a default source stream file location at the project level. This is set through the *Source path* field under the *Task defaults* section of the project properties.

This value is used when an application definition level contains a *Source path* reference of **\*PROJECT** and, similarly, when adding a new task for the project.

---

## User properties

The User properties within TURNOVER now allow you to specify a default IFS *Source path* under the *Programmer and resource* node of the user edit dialog if the user is enrolled as a programmer.



---

## Development

The **Checkouts** and **Object repository** filters under the *Development* node of the TURNOVER® Client now provides additional fields for source stream files. These new fields can be used in isolation or alongside other filter criteria that you choose to specify.

---

## Object stamping

By default, TURNOVER now enforces object stamping for objects that are created from source stream files (type code scheme of \*SRCSTMF). This is needed to ensure correct checkout, promotion and, in certain cases, reporting of source stream files within TURNOVER.

To ensure correct operation of TURNOVER, UNICOM strongly recommends that object stamping remains on for source stream file-created objects.

---

## Reports

A number of TURNOVER reports have been updated to cater for source stream files. These include:

- AUDITRPT (Audit report)
- AUDITADJ (Audit adjustments)
- LISTFORMS (List TURNOVER forms)
- RPTOBJHST (Object history report)
- RPTSRCUSE (Source checkout history)
- LISTPGMRS (List programmer profiles)

You now have all the information required to start using your new TURNOVER application definition for software changes that are underpinned by source stream files. The specified source path values will be respected throughout the software development lifecycle, for example when creating tasks, adding worklist items, running forms to promote changes, and distributing changes to remote systems, and so on.

## 6 Managing object types, schemes and data

This section describes how you can manage user-defined object types to selectively override change management methods that TURNOVER® Lifecycle Manager executes for an object.

### Overview

TURNOVER® Lifecycle Manager performs a significant amount of processing steps in order to correctly and safely promote the source and object for any given type code, including:

- How should the source be archived?
- How should its authority be changed?
- How should the original source be deleted?

These methods are collectively referred to as the TURNOVER® Lifecycle Manager *change management scheme* for that type code. TURNOVER® Lifecycle Manager always executed these methods using default system commands such as `MOV OBJ`, `CRTDUPOBJ`, `CHKOBJ`, and so on. (If the object type had a record in the TURNOVER® Lifecycle Manager User-defined Type file (TUSRTYPF), TURNOVER® Lifecycle Manager bypassed these methods. For more information about the TUSRTYPF file and its relationship to CM schemes, see [‘Relationship between the TUSRTYPF file and CM schemes’ on page 106.](#))

You can create your own change management schemes to define any of the sixteen *change management methods* that you would execute for an object type. For example, you might want a CM Scheme to:

- manage the promotion of certain data records in one of your company’s key database files *or*
- enable some enhanced promotion processing for SQL *or*
- use IBM’s CHGPF data conversion method instead of the more traditional methods such as `*MAP *DROP`.

Some example change management schemes are include with TURNOVER® Lifecycle Manager. You can use these schemes as examples or create your own schemes. See [‘Example change management schemes’ on page 111.](#)

### What you need to do

The steps you need to perform to set up and use a change management scheme are:

- 1 Create a change management scheme.

See *'Creating a CM scheme' on page 87*.

- 2 Override change management methods for the new change management scheme (as necessary).

See *'Overriding CM methods' on page 89*.

- 3 Create *change management method programs* for change management methods you have overridden.

See *'Create CM method programs' on page 96*.

- 4 Create new TURNOVER® Lifecycle Manager type codes or change existing TURNOVER® Lifecycle Manager type codes to use the new change management scheme.

See *'Example change management schemes' on page 111*.

## Creating a CM scheme

You must have TURNOVER® Lifecycle Manager System Defaults authority to create CM schemes.

- 1 On a TURNOVER® Lifecycle Manager command line, type TWRKCMSCH, and then press Enter.

The Work with Change Management Schemes panel appears.

|                                                         |                                         |                               |
|---------------------------------------------------------|-----------------------------------------|-------------------------------|
| MM/DD/YY<br>08:05:04                                    | Work with Change Management Schemes     | Your Company, Inc.<br>YOURSYS |
| 2=Change 3=Copy 4=Delete 5=Methods 7=Usage 9=Save       |                                         |                               |
| Scheme                                                  | Description                             |                               |
| ___ UTBIO                                               | Employee Biography Scheme               |                               |
| ___ UTDDG                                               | JDEdwards Data Dictionary Glossary Text |                               |
| ___ UTMENU                                              | Menu System Scheme                      |                               |
|                                                         |                                         | <u>Bottom</u>                 |
| F3=Exit F6=Add F8=Restore F12=Cancel F21=System command |                                         |                               |

This panel lists all the CM schemes that have been defined to TURNOVER® Lifecycle Manager. (For information about other options on this panel, see *'Overriding CM methods' on page 89*).

- 2 Press F6 to create a new scheme.

The Create CM Scheme panel appears.

```

                                Create CM Scheme (TCRTCMSCH)

Type choices, press Enter.

Scheme . . . . . Name
Description . . . . .
Default Program . . . . . *TURNOVER Name, *SCHEME, *NONE...
Library . . . . . *SCHEMELIB Name, *SCHEMELIB

                                Bottom
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

```

- 3 Specify the properties that are described below, and then press Enter.

TURNOVER® Lifecycle Manager creates the scheme and adds it to the list on the Work with Change Management Schemes panel.

### Scheme

A name to identify CM scheme.

### Description

A description of the CM scheme.

### Default Program

Each scheme comprises 16 individual methods. All methods are originally set to invoke the \*DEFAULT program in the \*DEFAULT library. These two parameters set the value of this scheme's default method processing program. For information about method programs and libraries, see '[Changing CM methods](#)' on page 90.

| Value       | Description                                                                                                                                                                                                                                                                                |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>name</i> | Type a default method program name.                                                                                                                                                                                                                                                        |
| *SCHEME     | The default method program is the same name as the scheme.                                                                                                                                                                                                                                 |
| *NONE       | For any method set to *DEFAULT, TURNOVER® Lifecycle Manager will perform no processing.<br><br>For example, if you are writing a scheme that handles objects with no source code, you might want to instruct TURNOVER® Lifecycle Manager to do no processing on the Archive Source method. |

| Value     | Description                                                                                                                                                                                                                                 |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| *TURNOVER | Use TURNOVER® Lifecycle Manager's standard processing methodology. This is useful if you need to customize the processing for only a few of the methods and all of the others you will defer back to TURNOVER® Lifecycle Manager's control. |
|           | For more information, see ' <a href="#">Overriding CM methods</a> ' on page 89.                                                                                                                                                             |

### Default Program Library

| Value       | Description                                                                     |
|-------------|---------------------------------------------------------------------------------|
| <i>name</i> | Type a default method library name.                                             |
| *SCHEMELIB  | TURNOVER® Lifecycle Manager gets the library name from the TSCHMELIB data area. |

## Overriding CM methods

To customize a CM scheme, you must override some of its CM methods to call your own processing programs. To override a CM method, you must change the name of the method program, the name of its library, or the names of both the program and library you want TURNOVER® Lifecycle Manager to call when executing the method.

### To customize a CM scheme

On the Work with Change Management Schemes panel, select the scheme with option 5.

The Work with Change Management Scheme Methods panel appears:

MM/DD/YY  
09:42:41

Work with Change Management Scheme Methods

Your Company, Inc.  
YOURSYS

Scheme . . . UTMENU      Menu System Scheme

Default method program . . . \*TURNOVER

Library . . . . . UTMENUPRD

2=Change

| Method       | Program  | Library  | Description              |
|--------------|----------|----------|--------------------------|
| — *ARCHIVE   | *DEFAULT | *DEFAULT | Archive Source           |
| — *BROWSE    | *DEFAULT | *DEFAULT | Browse Source            |
| — *CHGOBJ    | *DEFAULT | *DEFAULT | Change Object Attributes |
| — *CHGOBJOWN | *DEFAULT | *DEFAULT | Change Object Owner      |
| — *CHKOBJ    | *DEFAULT | *DEFAULT | Check for Object         |
| — *CHKSRC    | *DEFAULT | *DEFAULT | Check for Source         |
| — *CMPSRC    | *DEFAULT | *DEFAULT | Compare Source           |
| — *CPYSRCF   | *DEFAULT | *DEFAULT | Copy Source              |
| — *CRTDUPOBJ | *DEFAULT | *DEFAULT | Create Duplicate Object  |
| — *DLTOBJ    | *DEFAULT | *DEFAULT | Delete Object            |
| — *DLTSRC    | *DEFAULT | *DEFAULT | Delete Source            |

More...

F3=Exit    F11=Change scheme    F12=Cancel    F21=System command

- The panel lists the methods that can apply to a promotion situation. All methods are initially set to \*DEFAULT. This enables you to focus more on the “exceptions” than the “rule”. See [‘Changing CM methods’ on page 90](#).
- At the top of the panel is the current setting for this scheme’s default method processing program and library. Any method that lists \*DEFAULT for its processing program and/or library name will use these values. To change the values, press F11.

**Note** When you distribute application definitions with global type code definitions that reference change management schemes, TURNOVER® Lifecycle Manager does not send the change management scheme definitions and method programs. To send them to remote computers, use the TSAVCMSC (Save CM Scheme) and TRSTCMSC (Restore CM Scheme) commands. For more information, see [‘Saving a CM scheme’ on page 107](#) and [‘Restoring a CM scheme’ on page 108](#).

---

## Changing CM methods

---

### Method descriptions

---

The nature of the object type represented by the CM scheme determines if you need to override a CM method. For more information, see the [‘Notes’ on page 94](#). Also, see [‘Create CM method programs’ on page 96](#) and [‘Example change management schemes’ on page 111](#).)

#### **\*ARCHIVE (Archive Object’s Source)**

The default handing for this method is to copy the source member to the source file that is specified in the application definition and to compress the source member there.

You must provide a method program (or override this method to \*NONE) if the object type managed by this scheme is an object type that has source associated with it (that is, the global type code creation command references the “&SM”, “&SF”, and “&SL” substitution variables) but the “source” for the object type is not stored in the form of a physical file (source file or database file) member. (For example, the “source” might be keyed database records in several files that are processed by a “compiler” to generate some kind of object.)

The description about providing a method program also applies to other methods relating to source: \*CHKSRC, \*CMPSRC, \*CPYSRC, \*CPYSRCF, and \*DLTSRC. You can also override the \*CMPSRC method to \*NONE if the condition described above exists.

You do not have to provide a method program or override the \*ARCHIVE method to \*NONE if all applications that manage the object type handled by this scheme have \*NONE for the **Source archive file** application default.

This is one of several CM methods (the others are \*MISSING and \*RTVOBJD) that require that data (other than that returned by the **Return Error Flag**) be returned to the calling program. For more information, see [‘\\*ARCHIVE CM method: Archive object’s source’ on page 104](#).

### **\*BROWSE (Browse Object's Source)**

The default method handler for this method is the `STRSEU OPTION(5)` command.

You can provide a method program to browse objects of this object type if the source or the object itself (if the object type is non-source) for the object type managed by this scheme is not stored in a form that can be edited using `STRSEU` (source physical file).

### **\*CHGOBJ (Change Object Attributes)**

TURNOVER® Lifecycle Manager changes object attributes (based on production object, test object, reference object) based on object type.

You must provide a method program if you use the scheme to manage an object type other than those mentioned above, and if it is necessary to make some kind of change to the objects attributes upon promotion. If the object type is one of the types mentioned above, or if no changes need to be made to the object upon promotion, you can set this method to `*TURNOVER`.

### **\*CHGOBJOWN (Change Object Owner)**

The default method handler for this method is the `CHGOBJOWN` command.

You must provide a method program for this method if the object type managed by this scheme is not a valid value for the `CHGOBJOWN OBJTYPE` parameter.

If you do not need to change object ownership for the object type, you can override this method to `*NONE`.

### **\*CHKOBJ (Check for Object)**

The default method handler for this method is the `CHKOBJ` command.

You must override this method if the object type (global type code definition) represented by the scheme is not a valid value for the `CHKOBJ OBJTYPE` parameter. For example, if the object type is `*MYOBJTYPE`, you must override this method. If the object type is `*PGM`, you do not need to override this method.

### **\*CHKSRC (Check for Object's Source)**

The default method handler for this method is command `CHKOBJ` (with `OBJTYPE = *FILE` and `MBR = source member`).

For more information, see [\*\\*ARCHIVE \(Archive Object's Source\)' on page 90.\*](#)

### **\*CMPSRC (Compare Source Members for Object Type)**

TURNOVER® Lifecycle Manager uses the command found in the `TOCMPSRC` data area (shipped as `CMPSRCMBR`) as the default method handler for this method.

For more information, see [\*\\*ARCHIVE \(Archive Object's Source\)' on page 90.\*](#)

### **\*CPYSRCF (Copy object's source)**

The default method handler for this method is the `COPY FILE (CPYF)` command. This method includes checkout, promotion, copying archived source, and so on.

For more information, see *'\*ARCHIVE (Archive Object's Source)' on page 90* and *'CM method program considerations' on page 106*.

### **\*CRTDUPOBJ: Create Duplicate Object**

The default method handler for this method is the `CRTDUPOBJ` command.

You must override this method if the object type (global type code definition) represented by the scheme is not a valid value for the `CRTDUPOBJ OBJTYPE` parameter. Even if you have no intention of using TURNOVER® Lifecycle Manager promotion methods `CD` (Create Duplicate Object) or `CSCD` (Copy Source, Create Duplicate Object) with object types managed by the scheme, you must still provide a method program if the object type cannot be duplicated using the `CRTDUPOBJ` command. This is because TURNOVER® Lifecycle Manager uses `CRTDUPOBJ` to duplicate objects to a distribution work library when forms are distributed to other computers.

See also *'CM method program considerations' on page 106*.

### **\*DLTOBJ: Delete Object**

TURNOVER® Lifecycle Manager uses different commands to delete objects, depending on the object type. (The command to use for each object type is found in file `TDLTOBJF`.)

If the object type is a valid parameter for the `CHKOBJ OBJTYPE` parameter, it is probably unnecessary to provide a method program for this method. The type is probably already handled by TURNOVER® Lifecycle Manager's default handling.

### **\*DLTSRC: Delete Object's Source**

The default method handler for this method is the `RMVM` (Remove Member) command.

For more information, see *'\*ARCHIVE (Archive Object's Source)' on page 90*.

### **\*EDTSRC: Edit Object's Source**

The default method handler for this method is command `STRSEU OPTION(2)`, except when the Programmer Worklist Editor defaults are set to `*CODE400`.

While it is not required, you can provide a method program to edit this object type if the source or the object itself (if the object type is non-source) for the object type managed by this scheme is not stored in a form that can be edited using `STRSEU` (source physical file).

### **\*GRTOBJAUT: Grant Object Authorities**

The default method handler for this method is a combination of the `RVKOBJAUT`, `GRTOBJAUT`, and `CHGOBJPGP` commands. This method includes setting object authorities upon promotion, based on an existing object, reference object, and so forth.



If the object type managed by this scheme is not a valid value for the `GRTOBJAUT OBJTYPE` parameter, you must provide a method program for this method.

If it is not necessary to set authorities for the object type, you can override this method to `*NONE` i.

### **\*MISSING: Create Missing Objects for Object Type**

This is one of several CM methods (the others are `*ARCHIVE` and `*RTVOBJD`) that require that data (other than that returned by the **Return Error Flag**) be returned to the calling program.

For more information, see *'\*MISSING CM method: Create missing objects' on page 103*.

### **\*MOVOBJ: Move Object**

The default method handler for this method is the `MOVOBJ` command.

You must override this method if the object type (global type code definition) represented by the scheme is not a valid value for the `MOVOBJ OBJTYPE` parameter. Even if you have no intention of using TURNOVER® Lifecycle Manager promotion methods `MO` (Move Object) or `CSMO` (Copy Source, Move Object) with object types managed by the scheme, you must still provide a method program if the object type cannot be moved using the `MOVOBJ` command. This is because TURNOVER® Lifecycle Manager uses `MOVOBJ` to archive objects to the `TnnnnnnnD` library and to move them back from that library if TURNOVER® Lifecycle Manager recovers a form or runs a recovery form. (On a remote computer, TURNOVER® Lifecycle Manager also uses `MOVOBJ` to move objects to the `TnnnnnnnS` from-library when forms are received).

See also *'CM method program considerations' on page 106*.

### **\*RTVOBJD: Retrieve Object Description**

The default method handler for this method is the `RTVOBJD` command.

You might want to provide a method program for this method (but it is not required) if the object type represented by the scheme is not a valid value for the `RTVOBJD OBJTYPE` parameter. For example, you might provide a method program to retrieve the text description of objects so the text appears on the Programmer Worklist panel that displays the object description, and on several other TURNOVER® Lifecycle Manager panels that display object descriptions.

If the object type is not valid for `RTVOBJD` and a method program is not provided, override this method to `*NONE`. (TURNOVER® Lifecycle Manager does not then try to retrieve object descriptions for this scheme.)

This is one of several CM methods (the others are `*ARCHIVE` and `*MISSING`) that require that data (other than that returned by the **Return Error Flag**) be returned to the calling program.

See also *'\*RTVOBJD CM method: Retrieve object description' on page 101*.

## Notes

- Method programs that handle these methods must return Y as the **Return Error Flag** parameter if they cannot successfully execute the method (for example, if for \*ARCHIVE it cannot archive source or for \*CHGOBJ it cannot change the object).

For information about the **Return Error Flag**, see *'Create CM method programs' on page 96* and *'Error handling and messaging' on page 101*.

- The method program should send information messages to the calling program that describe why the action was unsuccessful. For example, a \*CHKSRC method program might return a message like "Definition for my special object MYOBJ not found in library MYDEVLIB" if the object's source does not exist. See *'Error handling and messaging' on page 101*.
- If the method program does not successfully execute methods \*CHGOBJ, \*CHGOBJOWN, and \*GRTOBJAUT, the TURNOVER® Lifecycle Manager form runs with warnings.

## Where the methods are used in TURNOVER® Lifecycle Manager

The following table summarizes where each CM method is used in TURNOVER® Lifecycle Manager.

| Method                              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| *ARCHIVE<br>Archive Object's Source | Archives a source member during a form promotion.<br><br>Applies only if the global type code involves source (that is, the global type code creation command references the "&SM", "&SF", and "&SL" substitution variables).                                                                                                                                                                                                                                                                                                                             |
| *BROWSE<br>Browse Object's Source   | <ul style="list-style-type: none"> <li>Browse an object description by pressing F2 on the Display Object Description (TDSPOBJD) panel. F2 is available if the object is an object type that has source, and the source can be found.</li> <li>Browse source on the Work with Object History panel, using option 1 (Browse current source) and option 2 (Browse archived source).</li> <li>Browse source on a worklist, using option 35.</li> <li>Browse source anywhere else that TURNOVER® Lifecycle Manager includes an option for doing so.</li> </ul> |
| *CHGOBJ<br>Change Object Attributes | Set object attributes during a form promotion.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| *CHGOBJOWN<br>Change Object Owner   | <ul style="list-style-type: none"> <li>Set object ownership (in primary and explode libraries) during a form promotion.</li> <li>Set object ownership in temporary distribution libraries.</li> </ul>                                                                                                                                                                                                                                                                                                                                                     |

| Method                    | Description                                                                                                                                                                                                                                                                                                                                                                                             |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| *CHKOBJ                   | <ul style="list-style-type: none"> <li>Set the <b>Obj Y/N</b> flag on a worklist.</li> </ul>                                                                                                                                                                                                                                                                                                            |
| Check for Object          | <ul style="list-style-type: none"> <li>Error checking when the promotion method is <b>MO</b> (Move Object), <b>CD</b> (Create Duplicate Object), <b>CSMO</b> (Copy Source, Move Object), or <b>CSCD</b> (Copy Source, Create Duplicate Object).</li> <li>Check for an object anywhere that TURNOVER® Lifecycle Manager needs to determine if an object exists. (There are many such places.)</li> </ul> |
| *CHKSRC                   | <ul style="list-style-type: none"> <li>Set the <b>Src Y/N</b> flag on a worklist.</li> </ul>                                                                                                                                                                                                                                                                                                            |
| Check for Object's Source | <ul style="list-style-type: none"> <li>Error checking when the promotion method is <b>CS</b> (Copy Source), <b>CSMO</b> (Copy Source, Move Object), or <b>CSCD</b> (Copy Source, Create Duplicate Object).</li> <li>Check for an object's source anywhere that TURNOVER® Lifecycle Manager needs to determine if source exists. (There are many such places.)</li> </ul>                                |
| *CPYSRCF                  | <ul style="list-style-type: none"> <li>Copying source during checkout.</li> </ul>                                                                                                                                                                                                                                                                                                                       |
| Copy Object's Source      | <ul style="list-style-type: none"> <li>Copying source during a form promotion.</li> <li>Copying source when changing a checkout record.</li> <li>Copying source to the distribution library.</li> <li>Copying source during receive of distribution.</li> </ul>                                                                                                                                         |
| *CRTDUPOBJ                | <ul style="list-style-type: none"> <li>Duplicating an object that has a non-source object type during checkout.</li> </ul>                                                                                                                                                                                                                                                                              |
| Create Duplicate Object   | <ul style="list-style-type: none"> <li>Duplicating an object during a form promotion.</li> <li>Duplicating an object to the distribution library.</li> <li>Duplicating an object to the explode library.</li> </ul>                                                                                                                                                                                     |
| *DLTOBJ                   | <ul style="list-style-type: none"> <li>Deleting an object during a form recovery.</li> </ul>                                                                                                                                                                                                                                                                                                            |
| Delete Object             | <ul style="list-style-type: none"> <li>Deleting a from-object during a form promotion</li> <li>Deleting an existing object before a worklist compile.</li> </ul>                                                                                                                                                                                                                                        |
| *DLTSRC                   | <ul style="list-style-type: none"> <li>Deleting source when changing a checkout record.</li> </ul>                                                                                                                                                                                                                                                                                                      |
| Delete Object's Source    | <ul style="list-style-type: none"> <li>Deleting source during a form recovery.</li> <li>Deleting a from-source during a form promotion.</li> <li>Deleting source for a delete line ("D") on a form.</li> </ul>                                                                                                                                                                                          |

| Method                                             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|----------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| *EDTSRC<br>Edit Object's Source                    | Editing source on a worklist, using option 32 (Edit source).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| *GRTOBJAUT<br>Grant Object Authorities             | Setting object authorities (in primary and explode libraries) during a form promotion.                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| *MISSING<br>Create Missing Objects for Object Type | Creating missing items on the Work with Application Definitions panel, using option 14 (Create missing items).                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| *MOVOBJ<br>Move Object                             | <ul style="list-style-type: none"> <li>▪ Moving out an original object during a form promotion.</li> <li>▪ Moving back an original object during a form recovery.</li> <li>▪ Moving an object from the <i>TnnnnnnnnS</i> library to the <i>Thhmmssjjj</i> library during distribution.</li> <li>▪ Moving an object during a form promotion, using method MO (Move Object) or CSMO (Copy Source, Move Object).</li> <li>▪ Moving an object from the <i>Thhmmssjjj</i> library to the <i>TnnnnnnnnS</i> library during receive of distribution.</li> </ul> |
| *RTVOBJD<br>Retrieve Object Description            | Returning data in a number of instances. For more information, see <i>'*RTVOBJD: Retrieve Object Description' on page 93</i> .                                                                                                                                                                                                                                                                                                                                                                                                                           |

## Create CM method programs

This section describes the method program parameters, which are the same for all CM method programs.

All CM method programs must accept these parameters:

| Parameter   | Length | Data type | Description                                                                                                                   |
|-------------|--------|-----------|-------------------------------------------------------------------------------------------------------------------------------|
| Method name | 10     | Character | The name of the CM method. This is the first parameter for all method programs. See <i>'Method descriptions' on page 90</i> . |
| Object data | 2000   | Character | Information about the object for which you are executing the method. See <i>'Object data parameter' on page 97</i> .          |

| Parameter         | Length | Data type | Description                                                                                                                                                                                                                                                                                                                             |
|-------------------|--------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Return error flag | 1      | Character | Specifies whether or not TURNOVER® Lifecycle Manager needs to perform error handling for the method program and send informational messages to the calling program.<br><br>The method program must return Y if it cannot successfully execute the method. For more information, see <i>'Error handling and messaging' on page 101</i> . |
| Return data       | 2000   | Character | Information returned by the method program. While only three CM methods (*ARCHIVE, *MISSING, and *RTVOBJD) require data to be passed back to the calling program, all method programs must accept this parameter. See <i>'Return data parameter' on page 101</i> .                                                                      |

## Object data parameter

The **Object data** parameter is the second parameter for all CM method programs. This parameter has 2000 positions, and not all of its data elements are relevant to all of the CM methods.

**Note** A method program should not depend on, for a particular CM method, data elements that are not listed as being relevant to that method.

| Data element          | Length | Data type      | Relevant CM methods |
|-----------------------|--------|----------------|---------------------|
| Form number           | 7      | Packed decimal | None                |
| Line number           | 3      | Packed decimal | None                |
| Object type code      | 10     | Character      | All                 |
| Object type           | 10     | Character      | All                 |
| Promotion method      | 5      | Character      | None                |
| Target object         | 10     | Character      | All                 |
| Target object library | 10     | Character      | All                 |

| Data element                       | Length | Data type | Relevant CM methods                                             |
|------------------------------------|--------|-----------|-----------------------------------------------------------------|
| Target source file                 | 10     | Character | *CHKSRC, *CPYSRCF, *DLTSRC, *EDTSRC, *BROWSE, *CMPSRC, *ARCHIVE |
| Target source library              | 10     | Character | *CHKSRC, *CPYSRCF, *DLTSRC, *EDTSRC, *BROWSE, *CMPSRC, *ARCHIVE |
| Target source member               | 10     | Character | *CHKSRC, *CPYSRCF, *DLTSRC, *EDTSRC, *BROWSE, *CMPSRC, *ARCHIVE |
| From object                        | 10     | Character | *MOVOBJ, *CRTDUPOBJ                                             |
| From object library                | 10     | Character | *MOVOBJ, *CRTDUPOBJ                                             |
| From source file                   | 10     | Character | *CPYSRCF, *ARCHIVE                                              |
| From source file library           | 10     | Character | *CPYSRCF, *ARCHIVE                                              |
| From source member                 | 10     | Character | *CPYSRCF, *ARCHIVE                                              |
| Programmer                         | 10     | Character | None                                                            |
| Reference                          | 10     | Character | None                                                            |
| Create parm                        | 1      | Character | *CHGOBJ                                                         |
| Create reference object            | 10     | Character | *CHGOBJ                                                         |
| Create reference object library    | 10     | Character | *CHGOBJ                                                         |
| Authority parm                     | 1      | Character | *CHGOBJOWN, *GRTOBJAUT                                          |
| Authority reference object         | 10     | Character | *CHGOBJOWN, *GRTOBJAUT                                          |
| Authority reference object library | 10     | Character | *CHGOBJOWN, *GRTOBJAUT                                          |
| Owner                              | 10     | Character | *CHGOBJOWN                                                      |



| Value  | Description                           |
|--------|---------------------------------------|
| *LINE  | Line number                           |
| *TYPED | Type code                             |
| *OBJTY | Object type                           |
| *HOWTO | Promotion method: CSCO, MO, and so on |
| *OBJ   | Target object                         |
| *LIB   | Target library                        |
| *TSRCF | Target source file                    |
| *TSRCL | Target source library                 |
| *TSRCM | Target source member                  |
| *FOBJ  | From object                           |
| *FLIB  | From library                          |
| *FSRCF | From source file                      |
| *FSRCL | From source library                   |
| *FSRCM | From source member                    |
| *PGMR  | Programmer                            |
| *REF   | Project and task reference            |
| *DFTS  | Create parameter: R, P, S, T          |
| *DFTSO | Create reference object               |
| *DFTSL | Create reference object library       |
| *AUTH  | Authority parameter: R, P, S, T       |
| *AUTHO | Authority reference object            |
| *AUTHL | Authority reference object library    |
| *OWNER | Owner                                 |

#### CL variable for returned value

Length: 10



Data type: Character

## TRTVCMPARM command example

A \*MOVOBJ method program for a user-defined object type might include this section of code:

```
TRTVCMPARM OBJDTA (&OBJDTA) PARMNAME (*FOBJ) RTNVAR (&FROMOBJ)
TRTVCMPARM OBJDTA (&OBJDTA) PARMNAME (*FLIB) RTNVAR (&FROMLIB)
TRTVCMPARM OBJDTA (&OBJDTA) PARMNAME (*LIB) RTNVAR (&TOLIB)
MYMOVCMD OBJ (&FROMLIB/&FROMOBJ) TOLIB (&TOLIB)
```

For more examples, see *'Example change management schemes' on page 111*.

---

## Error handling and messaging

All method programs must return a Y in the **Return Error Flag** parameter if the program cannot successfully execute its associated method. (For example, a \*CHKOBJ method program should return Y if it cannot find the object.) This ensures that TURNOVER® Lifecycle Manager's processing continues appropriately.

You can (optionally) have method programs send information messages to the calling program (\*PRV) that describe exactly why the action was unsuccessful or that indicate authorization problems, and so forth. For example, you could have a \*CHKOBJ method program return a message like "My special MYLIB/MYOBJ not found" if the object it is checking does not exist.

---

## Return data parameter

While all method programs must accept returned data (in the **Return data** parameter), these CM methods require the method program to return data (other than the **Return Error Flag**) to TURNOVER® Lifecycle Manager: \*RTVOBJD, \*ARCHIVE, and \*MISSING.

The required format for the returned data depends on which of these methods the method program is handling.

---

### \*RTVOBJD CM method: Retrieve object description

| Data element                  | Length | Data type |
|-------------------------------|--------|-----------|
| Object text                   | 50     | Character |
| Owner                         | 10     | Character |
| Return library                | 10     | Character |
| Object attribute              | 10     | Character |
| Object user-defined attribute | 10     | Character |
| Create date                   | 13     | Character |

| Data element        | Length | Data type |
|---------------------|--------|-----------|
| Source file         | 10     | Character |
| Source file library | 10     | Character |
| Source member       | 10     | Character |
| Source date         | 13     | Character |
| Object level        | 8      | Character |
| Licensed program    | 16     | Character |
| PTF                 | 10     | Character |
| APAR                | 10     | Character |

TURNOVER® Lifecycle Manager provides a Return Object Description Data file (TRTNOBJDTA), which has the field layout described in the previous table. This file is provided for your convenience; you can have CM method programs use it to parse the returned data using an externally-defined data structure. (See *'Example change management schemes' on page 111.*)

## TRTNOBJDTA command: Set RTNOBJDTA parameter

You can use the TRTNOBJDTA command in a CM method program (CL program only) to parse the return data.

**Set RTNOBJDTA Parameter (TRTNOBJDTA)**

Type choices, press Enter.

RTNOBJDTA in . . . . . \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Data type . . . . . \_\_\_\_\_ \*TEXT, \*OWNER, \*RTNLIB...

Value . . . . . \_\_\_\_\_ \*NULL

\_\_\_\_\_

CL variable for returned value \_\_\_\_\_ Character value

**Bottom**

F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display  
F24=More keys

## TRTNOBJDTA command example

A \*RTVOBJD method program for setting text and source data elements of the returned data might include this section of code (it is not necessary to set *all* elements for *all* object types).

```
TRTNOBJDTA OBJDTA (&RTNOBJDTA) DTATYP (*TEXT) VALUE (&TEXT) +
RTNOBJDTA (&RTNOBJDTA)
```

```
TRTNOBJDTA OBJDTA (&RTNOBJDTA) DTATYP (*SRCF) VALUE (&SRCF) +
RTNOBJDTA (&RTNOBJDTA)
```

```
TRTNOBJDTA OBJDTA (&RTNOBJDTA) DTATYP (*SRCFLIB) VALUE (&SRCFLIB) +
RTNOBJDTA (&RTNOBJDTA)
```

```
TRTNOBJDTA OBJDTA (&RTNOBJDTA) DTATYP (*SRCMBR) VALUE (&SRCMBR) +
RTNOBJDTA (&RTNOBJDTA)
```

For more examples, see *'Example change management schemes' on page 111*.

---

## \*MISSING CM method: Create missing objects

TURNOVER® Lifecycle Manager uses the \*MISSING CM method when it checks an application or application level for missing items. When it checks for missing objects, it examines all the type code definitions for the application or level.

The default handling of the \*MISSING CM method is to check, in the type code definition, for these objects for each type code:

- Libraries that are referenced as either the From or Target Source file library or From or Target object library.
- Source files that are referenced as either the From or Target Source file.
- Reference objects that are referenced as either the Creation or Authority Reference object.

For each of the above objects it does not find, TURNOVER® Lifecycle Manager returns a creation command (CRTLIB..., CRTSRCPF..., CRTREFOBJ...) for the missing object. TURNOVER® Lifecycle Manager displays the creation command in a panel selection list and executes the command if the user selects it.

You must override the \*MISSING CM method to \*NONE or provide a method program that performs the appropriate checks for the existence of the objects listed above and returns the appropriate creation command if they are not found, if any of these conditions exist:

- The source files do not actually represent IBM i source files.
- The libraries referenced in the type code do not actually represent IBM i libraries.
- The reference objects are not of an actual IBM i object type.

The data elements from the **Object data** parameter (see *'Object data parameter' on page 97*) that are significant for the \*MISSING CM method are:

| Data element  | Description                                                                              |
|---------------|------------------------------------------------------------------------------------------|
| Target object | The name of the library, source file, or reference object the method program will check. |

---

| Data element          | Description                                                                   |
|-----------------------|-------------------------------------------------------------------------------|
| Target object library | The library of the target object (library, source file, or reference object). |
| Object type           | LIBRARY, SRC-PF, or REFOBJ.                                                   |

Using the `TRTVCMPARM` command in a CL program, you can parse the target object, target object library, and object type data elements from the **Object data** parameter, as follows:

```
TRTVCMPARM OBJDTA (&OBJDTA) PARMNAME (*OBJ) RTNVAR (&OBJ)
```

```
TRTVCMPARM OBJDTA (&OBJDTA) PARMNAME (*LIB) RTNVAR (&LIB)
```

```
TRTVCMPARM OBJDTA (&OBJDTA) PARMNAME (*OBJTY) RTNVAR (&OBJTY)
```

Alternatively, you can use the `TCMPARMS` (Parameter Template) file in an RPG program to parse the **Object data** parameter using an externally-defined data structure.

The CM method program must check for the existence of the object, using whatever means is appropriate for the object type. If the program does not find the object, it must use the **Return data** parameter (see *'Return data parameter' on page 101*) to return the command string **TURNOVER®** Lifecycle Manager will use to create the object. If the object does not need to be created, then you should have the program return blanks in the **Return data** parameter.

### \*ARCHIVE CM method: Archive object's source

**TURNOVER®** Lifecycle Manager uses an `*ARCHIVE` method program to copy the archived source from the "source file" in the form archive library (`TnnnnnnnD`) to the archive "source file."

These files are probably not actual source files. If they were, it would not be necessary to override this method. See *'Important information' on page 105*.

The data elements from the **Object data** parameter (see *'Object data parameter' on page 97*) that are significant for the `*ARCHIVE` CM method are:

| Data element        | Description                                                                                                         |
|---------------------|---------------------------------------------------------------------------------------------------------------------|
| From source file    | The "source file" from which the source must be copied. (This will be the target source file on the form line.)     |
| From source library | The <code>TnnnnnnnD</code> library.                                                                                 |
| From source member  | The source "member" from which the source must be copied. (This will be the target source member on the form line.) |
| Target source file  | The name of the source file specified as the <b>Source archive file</b> in the application definition.              |

| Data element          | Description                                                                                                |
|-----------------------|------------------------------------------------------------------------------------------------------------|
| Target source library | The name of the library specified as the <b>Source archive file library</b> in the application definition. |
| Target source member  | The Mnnnnnnnnnn alias generated by TURNOVER® Lifecycle Manager.                                            |

You can use the TRTVCMPPARM command in a CL program to parse the data elements listed above from the **Object data** parameter, as follows:

```
TRTVCMPPARM OBJDTA (&OBJDTA) PARMNAME (*FSRCF) RTNVAR (&FSRCF)
```

```
TRTVCMPPARM OBJDTA (&OBJDTA) PARMNAME (*FSRCL) RTNVAR (&FSRCL)
```

```
TRTVCMPPARM OBJDTA (&OBJDTA) PARMNAME (*FSRCM) RTNVAR (&FSRCM)
```

```
TRTVCMPPARM OBJDTA (&OBJDTA) PARMNAME (*TSRCF) RTNVAR (&TSRCF)
```

```
TRTVCMPPARM OBJDTA (&OBJDTA) PARMNAME (*TSRCL) RTNVAR (&TSRCL)
```

```
TRTVCMPPARM OBJDTA (&OBJDTA) PARMNAME (*TSRCM) RTNVAR (&TSRCM)
```

Alternatively, you can use the TCMPARMS (Parameter Template) file in an RPG program to parse the **Object data** parameter using an externally-defined data structure.

## Important information

If normal object types (that is, object types that do not use a CM scheme) are managed by the same application definitions that manage object types with schemes that override the \*ARCHIVE method, then the **Source archive file** specified in the application definition is probably an IBM i source file and, therefore, not an appropriate target for archiving “source” for this object type. The method program must archive the “source” to the appropriate place.

If the appropriate place is NOT the same as the Target source file, Target source file library, and Target source member data elements that were passed to the method program in the **Object data** parameter (see [“\\*ARCHIVE CM method: Archive object's source” on page 104](#)), then the method program must use the **Return data** parameter (see [“Return data parameter” on page 101](#)) to return the actual values that identify where the “source” was archived. The format of the returned data must be:

- Archive file (positions 1-10)
- Archive file library (positions 11-20)
- Archive member (positions 21-30).

TURNOVER® Lifecycle Manager uses the returned data to correctly update the TURNOVER® Lifecycle Manager archive history database. TURNOVER® Lifecycle Manager then knows where the archived source resides if a request is made to browse or copy it.

The sample CM method programs that are shipped with TURNOVER® Lifecycle Manager include an \*ARCHIVE method program that demonstrates this concept.

## CM method program considerations

This section presents additional information that you should be aware of when working with CM method programs.

### Establishing an object's environment (in a transient library)

There are several situations when TURNOVER® Lifecycle Manager creates libraries and copies or moves source and objects to those libraries. For example, TURNOVER® Lifecycle Manager creates `TnnnnnnnnD` archive libraries when forms are run. TURNOVER® Lifecycle Manager moves existing versions of objects that are on the form to that library before creating the new version. Also, if there is source for the object type, TURNOVER® Lifecycle Manager copies the existing version of that source to the `TnnnnnnnnD` archive library. When a form is distributed, TURNOVER® Lifecycle Manager creates a work library and duplicates objects into that library, which it then saves. When a distributed form is received, TURNOVER® Lifecycle Manager again moves and copies objects and source into the `TnnnnnnnnS` from libraries.

Consequently, when a scheme uses a method program to handle the `*MOVOBJ`, `*CRTDUPOBJ`, or `*CPYSRCF` methods, you must keep an important consideration in mind when writing the program. That is, the method program must be capable of moving or duplicating objects and copying source into an empty library. The program must do whatever is necessary to establish the target environment for the move or copy. For example, if records in one or more database files represent an object type, then the method program must automatically create those database files in the target library if they do not exist there already. For examples, see [‘Example change management schemes’ on page 111](#).

### Relationship between the TUSRTYPE file and CM schemes

The TUSRTYPEF (TURNOVER® Lifecycle Manager User-defined Type) file, serves these purposes:

- If TURNOVER® Lifecycle Manager finds a record in file TUSRTYPEF for the object type (for example, `*JDEDW`), it bypasses operations (such as changing owner, setting authorities, and so forth) that are not appropriate for the object type, when a form runs. By doing so, TURNOVER® Lifecycle Manager does not generate errors (for example, “`*JDEDW` not valid for parameter OBJTYPE”) that would result in form warnings.

**Note** Object types that you will manage using CM schemes should *not* have records in file TUSRTYPEF. To minimize impact on existing interfaces and rare instances of TURNOVER® Lifecycle Manager hard-coding to accommodate special object types, TUSRTYPEF continues to serve the same purpose as with previous releases (see [‘Relationship between the TUSRTYPEF file and CM schemes’ on page 106](#)) during the running of a form. If there is a record in file TUSRTYPEF, TURNOVER® Lifecycle Manager still bypasses operations such as changing authorities, setting owners, and so forth, even if the object type uses a CM scheme.

## Create type codes to use a CM scheme

Once you create a CM scheme and override CM methods (as necessary), you can create new TURNOVER® Lifecycle Manager type codes or change existing TURNOVER® Lifecycle Manager type codes to use the new scheme. For information about how to do this, see “Managing type code definitions” in the *TURNOVER® Lifecycle Manager User Guide*.

## Using CM schemes

### Displaying a CM scheme's usage

To display a list of TURNOVER® Lifecycle Manager global type codes that are currently using a CM scheme, select a scheme on the Work with Change Management Schemes panel with option 7.

| MM/DD/YY                                          |                           | Work with Change Management Schemes |  | Your Company, Inc. |               |
|---------------------------------------------------|---------------------------|-------------------------------------|--|--------------------|---------------|
| 15:48:24                                          |                           |                                     |  | YOURSYS            |               |
| 2=Change 3=Copy 4=Delete 5=Methods 7=Usage 9=Save |                           |                                     |  |                    |               |
| <b>Scheme</b>                                     | <b>Description</b>        |                                     |  |                    |               |
| UTBIO                                             | Employee Biography Scheme |                                     |  |                    |               |
| <u>7</u> UTMENU                                   | Menu System Scheme        |                                     |  |                    |               |
| Scheme . . . . . UTMENU                           |                           |                                     |  |                    |               |
| Description . . . . . Menu System Scheme          |                           |                                     |  |                    |               |
| Scheme is used by the following type codes:       |                           |                                     |  |                    |               |
| <b>Type</b>                                       | <b>Description</b>        |                                     |  |                    |               |
| UTMENU                                            | My Menu System Menu       |                                     |  |                    |               |
|                                                   |                           |                                     |  |                    | <b>Bottom</b> |
| F3=Exit F12=Cancel                                |                           |                                     |  |                    |               |

### Saving a CM scheme

You can save a CM scheme definition, and, optionally, its method programs and resource library, to a save file. (The resource library contains any additional objects required by the method programs.)

This command is useful for exporting CM schemes to other systems (see also [‘Restoring a CM scheme’ on page 108](#)). This is necessary if the application definition in which you have created this scheme and its associated type code also exists on remote (that is, live production) machines to which you are distributing this type code’s objects.

To save a CM scheme, do one of the following:

- On the Work with Change Management Schemes panel, select the scheme with option 9.
- On a command line, type TSAVCMSCH, and then press Enter.

```

                                Save CM Scheme (TSAVCMSCH)

Type choices, press Enter.

Scheme . . . . . > UTMENU      Name
Save method programs . . . . . *YES      *YES, *NO
Save resource library . . . . . *NONE      Name, *NONE
Save file . . . . .           Name
      Library . . . . . *LIBL      Name, *LIBL
Target release . . . . . *CURRENT      *CURRENT, *PRV, VnRnMn ...

                                Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys

```

### Save method programs

To also save the CM method programs the scheme uses, specify \*YES.

### Save resource library

To also save the scheme's resource library, specify its name.

The resource library contains any additional objects, such as files, called programs, commands, and so forth, required by the scheme's method programs.

### Save file, Save file Library

Specify the name of the save file and library to which you want to save the scheme.

To search for the save file you have specified using the current library list, specify \*LIBL.

If the save file does not exist, TURNOVER® Lifecycle Manager creates it.

### Target release

Specify the target system's IBM i release level you want TURNOVER® Lifecycle Manager to use for the save commands (SAVLIB, SAVOBJ, and so on) when it saves the scheme, method programs, and resource library.

---

## Restoring a CM scheme

You can restore a previously saved CM scheme definition and its method programs and resource library, from a save file that was created with the TSAVCMSCH command.

To restore a saved CM scheme, do one of the following:

- On the Work with Change Management Schemes panel, press F8.
- On a command line, type TRSTCMSCH, and then press Enter).



The Restore CM Scheme panel appears.

**Restore CM Scheme (TRSTCMSCH)**

Type choices, press Enter.

|                                       |            |                  |
|---------------------------------------|------------|------------------|
| Save file . . . . .                   | *LIBL      | Name             |
| Library . . . . .                     | *LIBL      | Name, *LIBL      |
| Restore to scheme . . . . .           | *SAVSCHEME | Name, *SAVSCHEME |
| Replace existing scheme . . . . .     | *NO        | *NO, *YES        |
| Restore method programs . . . . .     | *YES       | *YES, *NO        |
| RSTLIB for method programs . . . . .  | *SCHEME    | Name, *SCHEME    |
| Restore resource library . . . . .    | *YES       | *YES, *NO        |
| RSTLIB for resource library . . . . . | *SAVLIB    | Name, *SAVLIB    |

**Bottom**

F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display  
F24=More keys

### Save file, Save file library

Specify the name of the save file and library from which you want TURNOVER® Lifecycle Manager to restore the scheme. The save file you specify must have been created with the `TSAVCMSCH` command; see [‘Saving a CM scheme’ on page 107](#).

To search for the save file you have specified using the current library list, specify `*LIBL`.

### Restore to scheme

Specify the name of the scheme you want to create or update.

If the name of the scheme you want to restore is the same as the scheme that was saved, specify `*SAVSCHEME`.

### Replace existing scheme

If a scheme with the same name already exists:

- `*YES` replaces it with the scheme that you are restoring.
- `*NO` cancels the command.

### Restore method programs

To also restore any CM method programs that were saved with the scheme, specify `*YES`.

### RSTLIB for method programs

If you specify `*YES` for **Restore method programs**: Specify the name of the library to which you want to restore the CM method programs.

To use the library that is specified for the scheme method, specify `*SCHEME`.

## Restore resource library

To restore any resource library that was saved with the scheme, specify \*YES.

## RSTLIB for resource library

If you specify \*YES for **Restore resource library**: Specify the name of the library to which you want to restore the resource library.

To restore the resource library from the same library to which it was saved, specify \*SAVLIB.

## Generating source from a CM scheme

Use the TGENSCHSRC command to generate a shell or “starter” CL source member for any (or all) of the sixteen methods. This shell source code will have all of the variables defined, you just fill in the necessary logic in the middle.

| Restore CM Scheme (TRSTCMSCH)                                       |            |                  |
|---------------------------------------------------------------------|------------|------------------|
| Type choices, press Enter.                                          |            |                  |
| Save file . . . . .                                                 | _____      | Name             |
| Library . . . . .                                                   | *LIBL      | Name, *LIBL      |
| Restore to scheme . . . . .                                         | *SAVSCHEME | Name, *SAVSCHEME |
| Replace existing scheme . . . .                                     | *NO        | *NO, *YES        |
| Restore method programs . . . .                                     | *YES       | *YES, *NO        |
| RSTLIB for method programs . . .                                    | *SCHEME    | Name, *SCHEME    |
| Restore resource library . . . .                                    | *YES       | *YES, *NO        |
| RSTLIB for resource library . .                                     | *SAVLIB    | Name, *SAVLIB    |
| <b>Bottom</b>                                                       |            |                  |
| F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display |            |                  |
| F24=More keys                                                       |            |                  |

## Scheme

Specify the scheme for which you want to generate source.

## Method

Supply the name of the method for which you would want to generate source.

## Source File, Library

Specify the name of the source file and library to contain the generated source code.

## Replace or Add

To replace existing source code member's source, specify \*REPLACE.

To create a new source member, specify \*ADD.

## Example change management schemes

### Example 1: UTMENU scheme (menu system)

The UTMENU CM scheme manages changes to menus in a menu system that stores a menu's run-time definition in three files. The files and their fields are:

#### UTMENU: Menu header file

| Field                    | Description                                                  |
|--------------------------|--------------------------------------------------------------|
| Menu name                | Key                                                          |
| Menu description         |                                                              |
| Menu owner               | Retrieved using *RTVOBJD method; set using *CHBOBJOWN method |
| Menu active              | Y/N (Set using *CHGOBJ method)                               |
| Menu source file         | Retrieved using *RTVOBJD method; set by CRTUTMENU command    |
| Menu source file library | Retrieved using *RTVOBJD method; set by CRTUTMENU command    |
| Menu source member       | Retrieved using *RTVOBJD method; set by CRTUTMENU command    |
| Menu create date         | Retrieved using *RTVOBJD method; set by CRTUTMENU command    |
| Menu source date         | Not used                                                     |

#### UTMENUOPT: Menu options file

| Field                     | Description |
|---------------------------|-------------|
| Menu name                 | Key         |
| Option name               | Key         |
| Option description        |             |
| Option command to execute |             |

#### UTMENUAUT: Menu authorized users file

| Field     | Description |
|-----------|-------------|
| Menu name | Key         |

| Field           | Description |
|-----------------|-------------|
| Authorized user | Key         |

The UTMENAUT file is populated by the CRTUTMENU command, based on the <AUTHORITYLIST> tag in the XML source. Depending on the application definition, the UTMENAUT file gets updated by the \*GRTOBJAUT CM method.

In addition to the UTMENU, UTMENUOPT, and UTMENUAUT files, where a menu's run-time definition is stored, another file, UTMENSRC, stores XML style "source" (the file is not actually a source physical file) members that TURNOVER® Lifecycle Manager uses to construct menus using the UTCRTMENU command.

The UTCRTMENU command parses the XML source definition for the menu, writing records to the UTMENU, UTMENUOPT and UTMENAUT files to create the menu definition.

**Note** This example illustrates the usage of a CM scheme, and not how to design a menu system or an XML compiler. Consequently, it does not include any commands or programs to actually run a menu once the definition has been created. Also, the UTCRTMENU command imposes restrictions on how the XML document (UTMENSRC member) must be formatted beyond those that would be imposed by a true XML parser (see below).

The restrictions imposed by the UTCRTMENU command include these:

- Tag names must be uppercase.
- <DESCRIPTION> and </DESCRIPTION> must be on the same line.
- <COMMAND> and </COMMAND> must be on the same line.
- <OWNER> and </OWNER> must be on the same line.
- <ACTIVE> and </ACTIVE> must be on the same line.
- Anything else on the same line as <OPTIONS> or </OPTIONS> is ignored.
- Anything else on the same line as <OPTION name="xxxxxxxxxx"> or </OPTION> is ignored.
- Anything else on the same line as <AUTHORITYLIST> or </AUTHORITYLIST> is ignored.

The UTMENU CM scheme is contrived. It uses an overridden CM method program for every possible method, which would rarely be required for a CM scheme. More likely, at least some of the methods would be overridden to \*NONE.

## Restoring the UTMENU scheme

- 1 Sign on as a user profile with authority to restore libraries and TURNOVER® Lifecycle Manager application definitions to the system.
- 2 On a TURNOVER® Lifecycle Manager command line, type TWRKCMSCH, and then press Enter.

The Work with Change Management Schemes panel appears.

| Work with Change Management Schemes                     |                                         | Your Company, Inc.<br>YOURSYS |
|---------------------------------------------------------|-----------------------------------------|-------------------------------|
| MM/DD/YY<br>08:05:04                                    |                                         |                               |
| 2=Change 3=Copy 4=Delete 5=Methods 7=Usage 9=Save       |                                         |                               |
| Scheme                                                  | Description                             |                               |
| UTBIO                                                   | Employee Biography Scheme               |                               |
| UTDDG                                                   | JDEdwards Data Dictionary Glossary Text |                               |
| UTMENU                                                  | Menu System Scheme                      |                               |
|                                                         |                                         | <b>Bottom</b>                 |
| F3=Exit F6=Add F8=Restore F12=Cancel F21=System command |                                         |                               |

- 3 If the UTMENU scheme is listed on this panel, go to step 6.
- 4 Press F8 (Restore).

The Restore CM Scheme (TRSTCMSCH) panel appears.

| Restore CM Scheme (TRSTCMSCH)                                                        |                   |                  |
|--------------------------------------------------------------------------------------|-------------------|------------------|
| Type choices, press Enter.                                                           |                   |                  |
| Save file . . . . .                                                                  |                   | Name             |
| Library . . . . .                                                                    | <u>*LIBL</u>      | Name, *LIBL      |
| Restore to scheme . . . . .                                                          | <u>*SAVSCHEME</u> | Name, *SAVSCHEME |
| Replace existing scheme . . . .                                                      | <u>*NO</u>        | *NO, *YES        |
| Restore method programs . . . .                                                      | <u>*YES</u>       | *YES, *NO        |
| RSTLIB for method programs . . .                                                     | <u>*SCHEME</u>    | Name, *SCHEME    |
| Restore resource library . . . .                                                     | <u>*YES</u>       | *YES, *NO        |
| RSTLIB for resource library . .                                                      | <u>*SAVLIB</u>    | Name, *SAVLIB    |
|                                                                                      |                   | <b>Bottom</b>    |
| F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display<br>F24=More keys |                   |                  |

- 5 Type UTMENUSAVF for the save file name. Accept the default values for the other parameters (the save file is in the TURNOVER® Lifecycle Manager product library). Press Enter.

TURNOVER® Lifecycle Manager restores these items:

- the UTMENU scheme definition.
- library UTMENUPRD (the resource library), which contains the menu definition and menu source files, the UTCRTMENU command (and command processing programs), and the UTMENU CM method programs.

- 6 On the main menu, select option 1 (Work with Application Definitions).

If application UT is already on the Work with Application Definitions panel (you might have to page down to see it), proceed to step 8.

- 7 On a TURNOVER® Lifecycle Manager command line, type TRSTAPP.

The Restore Application (TRSTAPP) panel appears.

| Restore Application (TRSTAPP)                                       |               |                 |  |
|---------------------------------------------------------------------|---------------|-----------------|--|
| Type choices, press Enter.                                          |               |                 |  |
| Save file . . . . .                                                 |               | Name            |  |
| Library . . . . .                                                   |               | Name            |  |
| Restore to Application . . . . .                                    | <u>*SAV</u>   | Character value |  |
| Release . . . . .                                                   | _____         | Number          |  |
| Version . . . . .                                                   | _____         | Number          |  |
| Restore option . . . . .                                            | <u>*BASIC</u> | *BASIC, *FULL   |  |
| <b>Bottom</b>                                                       |               |                 |  |
| F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display |               |                 |  |
| F24=More keys                                                       |               |                 |  |

- 8 Type UTAPPSAVF for the save file name. For the save file library, type the name of your TURNOVER® Lifecycle Manager product library (normally SOFTTURN).

| Restore Application (TRSTAPP)                                       |               |                 |  |
|---------------------------------------------------------------------|---------------|-----------------|--|
| Type choices, press Enter.                                          |               |                 |  |
| Save file . . . . .                                                 | UTAPPSAVF     | Name            |  |
| Library . . . . .                                                   | SOFTTURN      | Name            |  |
| Restore to Application . . . . .                                    | <u>*SAV</u>   | Character value |  |
| Release . . . . .                                                   | _____         | Number          |  |
| Version . . . . .                                                   | _____         | Number          |  |
| Restore option . . . . .                                            | <u>*BASIC</u> | *BASIC, *FULL   |  |
| <b>Bottom</b>                                                       |               |                 |  |
| F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display |               |                 |  |
| F24=More keys                                                       |               |                 |  |

- 9 Press Enter.

TURNOVER® Lifecycle Manager restores the UT application definition as well as the UTMENU global type code definition.

- 10 On the main menu, select option 8 (Utility Menu) and then option 4 (Work with type code definitions).

The Work with Type Codes panel appears.

11 Type 2 next to the UTMENU type code.

| MM/DD/YY                 |                          | Work with Type Codes    |        | Your Company, Inc.    |                   |
|--------------------------|--------------------------|-------------------------|--------|-----------------------|-------------------|
| 16:03:02                 |                          |                         |        | YOURSYS               |                   |
|                          |                          | Position to . . . _____ |        |                       |                   |
| 2=Change 3=Copy 4=Delete |                          |                         |        |                       |                   |
| Type                     | Description              | Object Type             | Object | Seq                   | Create CMD        |
| __ TBL                   | Translation Table        | *TBL                    | N      | 400                   | CRTTBL TBL("&LI"/ |
| __ TXT                   | Copy Text Member         | *SRC                    | N      | 100                   |                   |
| __ T3                    | AS/400 Physical File     | *FILE                   | Y      | 410                   | CRTPF FILE("&LI"/ |
| __ UIMMNU                | UIM Menu                 | *MENU                   | N      | 200                   | CRTMNU MENU("&LI" |
| __ UTBIO                 | User-defined Biography   | *UTBIO                  | N      | 100                   |                   |
| 2 __ UTMENU              | My Menu System Menu      | *UTMENU                 | N      | 100                   | UTCRTMENU MENU("& |
| __ WORD                  | Word Documents           | *IFSOBJ                 | Y      |                       |                   |
| __ YINT                  | Synon/2E Non-source type | *SGT                    | N      | 900                   |                   |
| __ YUFC                  | User Source (COBOL)      | *SRC                    | N      | 100                   |                   |
| __ YUFR                  | User source              | *SRC                    | N      | 100                   |                   |
| __ YYCLP                 | CL Program               | *PGM                    | N      | 500                   | &PRDLIB/TCRTOBJY  |
| __ YYCMD                 | Command                  | *CMD                    | N      | 400                   | &PRDLIB/TCRTOBJY  |
| __ YYDSPF                | AS/400 Display File      | *FILE                   | N      | 450                   | &PRDLIB/TCRTOBJY  |
| __ YYLF                  | AS/400 Logical File      | *FILE                   | Y      | 425                   | &PRDLIB/TCRTOBJY  |
| __ YYPF                  | AS/400 Physical File     | *FILE                   | Y      | 410                   | &PRDLIB/TCRTOBJY  |
| More...                  |                          |                         |        |                       |                   |
| F3=Exit                  |                          | F6=Add Type             |        | F8=User-defined parms |                   |
|                          |                          |                         |        | F24=More Keys         |                   |

12 Press Enter.

13 Make sure that the type code definition you now see matches this:

| MM/DD/YY              |                                                                | Change a Type Code                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |          | Your Company, Inc.                                |  |
|-----------------------|----------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------------------------------------------|--|
| 16:05:04              |                                                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |          | YOURSYS                                           |  |
| Type code . . . . .   | UTMENU                                                         | CM scheme . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | UTMENU   |                                                   |  |
| Description . . . . . | My Menu System Menu                                            | Attribute . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |          |                                                   |  |
| Object type . . . . . | *UTMENU                                                        | Source file name . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | UTMENSRC |                                                   |  |
| Sequence . . . . .    | 100                                                            | Data object . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | N        |                                                   |  |
| Create command . . .  | UTCRTMENU MENU("&LI"/"&OB") SRCFILE("&SL"/"&SF") SRCMBR("&SM") |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |          |                                                   |  |
|                       |                                                                | "&RF" = Reference      "&X0"-"&X9" = Exit<br>"&OB" = Object name      "&TO" = Test object name<br>"&LI" = Library name      "&TL" = Test object library name<br>"&TY" = Type code      "&TR" = Target release<br>"&SM" = Source member name      "&FM" = From source member name<br>"&SF" = Source file name      "&FF" = From source file name<br>"&SL" = Source library name      "&FL" = From source library name<br>"&U0" = Generation options      "&U1" = Debugging views<br>"&U2" = Optimization level      "&U3" = Compiler options<br>"&U4" = Source listing options      "&U5" = Unassigned<br>"&U6" = Unassigned      "&U7" = Service program source file<br>"&U8" = Binding directory      "&U9" = Module source file |          |                                                   |  |
| F3=Exit               |                                                                | F4=Prompt/select                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |          | F7=Applications    F12=Cancel    F22=Long command |  |

14 Return to the Work with Application Definitions panel, and then type option 14 (Create missing items) next to the UT application.

Unless they have been created previously, you will be prompted to create library UTMENUDEV (development library) and file UTMENUDEV/UTMENSRC (development source). Because UTMENSRC is not actually a source file, we have used the UTMENU CM scheme (\*MISSING method) to override (from CRTPFxxxx to CRTDUPOBJxxxxx) the create command for the source file.

15 Press Enter to create the missing objects.

Now you can create a Programmer Worklist and start experimenting with the UTMENU CM scheme. To start with, you can work with two existing menu definitions: ACCOUNTING and TESTMENU.

## UTMENU scheme objects

The source code for all the objects that were restored into library UTMENUPRD when you restored the UTMENU CM scheme are in source files ADDSSRC (PFs), ACMDSRC (CMD), ACLSRC (CLLEs) and ARPGLESRC (RPGLEs) in the TURNOVER® Lifecycle Manager product library. The restored objects include:

| Object                                       | Description                                                                                                                                                                                             |
|----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UTMENU, UTMENOPT, UTMENAUT, UTMENSRC         | The menu definition files described in <i>'Example 1: UTMENU scheme (menu system)' on page 111</i>                                                                                                      |
| UTMENARC                                     | The source archive file for the UTMENU scheme. The format is identical to UTMENSRC. (See the description of method program UTARCHIVE later in this section for a description of how this file is used.) |
| UTCRTMENU (CMD, CLLE) and UTCRTMENUR (RPGLE) | The command and command processing programs used to construct UTMENU menus                                                                                                                              |
| UTARCHIVE (CLLE)                             | *ARCHIVE method program                                                                                                                                                                                 |
| UTCHGOBJ (CLLE) and UTCHGOBJR (RPGLE)        | *CHGOBJ method program                                                                                                                                                                                  |
| UTCHGOWN (CLLE) and UTCHGOWNR (RPGLE)        | *CHGOBJOWN method program                                                                                                                                                                               |
| UTCHKOBJ (CLLE) and UTCHKOBJR (RPGLE)        | *CHKOBJ method program                                                                                                                                                                                  |
| UTCHKSRC (CLLE)                              | *CHKSRC method program                                                                                                                                                                                  |
| UTCMPSRC (CLLE)                              | *CMPSRC method program                                                                                                                                                                                  |
| UTCPYSRCF (CLLE)                             | *CPYSRCF method program                                                                                                                                                                                 |
| UTDLTOBJ (CLLE) and UTDLTBJR (RPGLE)         | *DLTOBJ method program                                                                                                                                                                                  |
| UTDLTSRC (CLLE)                              | *DLTSRC method program                                                                                                                                                                                  |
| UTEDTBRW (CLLE)                              | *EDTSRC and *BROWSE method program                                                                                                                                                                      |
| UTGRTAUT (CLLE) and UTGRTAUTR (RPGLE)        | *GRTOBJAUT method program                                                                                                                                                                               |



| Object                                  | Description                            |
|-----------------------------------------|----------------------------------------|
| UTMISSING (CLLE)                        | *MISSING method program                |
| UTMOVDUP (CLLE) and UTMOVDUPR (RPGLE)   | *MOVOBJ and *DRTDUPOBJ method programs |
| UTRTVOBJD (CLLE) and UTRTVOBJDR (RPGLE) | *RTVOBJD method program                |

## CM method program parameters

All CM method programs must accept the same four parameters (see [‘Create CM method programs’ on page 96](#)). You can use the parameter definitions for the UTMENU method programs as templates for creating your own method programs.

## Using the TRTVCMPARM command in a CL method program

The second parameter passed to all CM method programs is a 2000-character string from which all information needed to execute the method can be retrieved (object name, from library, target library, and so forth). You can use the TURNOVER® Lifecycle Manager command, TRTVCMPARM, to retrieve individual data elements from this 2000-character string. For examples of this, review the source code for any of the following UTMENU method programs:

UTARCHIVE, UTCHGOWN, UTCHKSR, UTMPSRC, UTMPSRCF, UTDLTBJ, UTDLTSC, UTDLTBRW, UTMISSING, UTMOVDUP, UTRTVOBJD

## Using TCMPARMS file for an externally-defined data structure in RPGLE

You can use the externally-defined TURNOVER® Lifecycle Manager file, TCMPARMS, in an RPGLE program to create a data structure over the second parameter passed to all CM method programs (see above) to parse the individual data elements from this 2000-character string. For examples of this technique, see the source code for any of the following UTMENU method programs (these programs are actually called by method programs):

UTCHGOBJR  
UTCHKOBJR  
UTGRTAUTR

## Using the TRTNOBJDTA command in a CL method program

The \*RTVOBJD (Retrieve Object Description) CM method program must return object description data using the fourth parameter, which is a 2000-character string that must be correctly formatted (that is, correct data elements in correct positions). You can use the TURNOVER® Lifecycle Manager command, TRTNOBJDTA, to set the individual data elements to the appropriate values. For an example of this, see the source code for the UTMENU method program UTRTVOBJD.

## Using TRTNOBJDTA file for an externally-defined data structure in RPGLE

You can use the externally-defined TURNOVER® Lifecycle Manager file, TRTNOBJDTA, in an RPGLE program to create a data structure over the fourth parameter (see previous paragraph) to easily parse the individual data elements from this 2000-character string. For examples of this, review the source code for the UTMENU method program UTCHGOBJR (called by method program UTCHGOBJ).

## Handling multiple CM methods in the same method program

When you are creating a CM scheme, you might choose to create separate method programs for each CM method, or to create method programs that handle more than one method. (A single method program could even handle all the methods.) For examples of method programs that handle more than one method, review the source code for the UTMENU method programs UTEDTBRW (\*EDTSRC and \*BROWSE methods) and UTMOVDUP (\*MOV OBJ and \*CRTDUPOBJ methods).

## Executing a CM method from another CM method

Sometimes you might find it useful to execute one CM method from within the method program that handles another CM method. You can use the TURNOVER® Lifecycle Manager command, TRUNCMMTH, to do this. For an example, review the source code for the UTMENU program UTMISsing (\*MISSING method), which uses the TRUNCMMTH command to execute the \*CHKOBJ (Check for Object) method.

## Overriding the source archive file from the \*ARCHIVE method

To archive source for an object type that does not store its source code in actual source physical files, you must provide a method program for the \*ARCHIVE method. Because the source archive file is a source physical file, the \*ARCHIVE method program will typically not archive to the source archive file specified in the application definition. (The source archive file is passed to the method program as one of the data elements of the second parameter.) In case the method program has to redirect the archived "source", the method program must return the actual target of the archived source in the fourth parameter, using this format:

- Archive file (positions 1-10)
- Archive file library (positions 11-20)
- Archive member (positions 21-30).

For an example, review the source code for the UTMENU method program UTARCHIVE.

## Checking for missing objects with the \*MISSING method

See the description of the \*MISSING (Create Missing Items) method (see [\\*MISSING CM method: Create missing objects' on page 103](#)), and then see the source code for the UTMENU method program UTMISsing as an example of how you use this method.

## Retrieving the default scheme library from the TSCHMELIB data area

When defining a CM scheme, one of the special values you can use for the method program library (for the scheme's default method program library or for an individual method's method program libraries, or for both) is \*SCHEMELIB. This special value causes TURNOVER® Lifecycle Manager to retrieve the library name from the TSCHMELIB data area. (The TURNOVER® Lifecycle Manager user or administrator is responsible for setting this data area to the appropriate library name). If a CM method program needs to call other programs, and you cannot assume that TURNOVER® Lifecycle Manager will find those programs in the library list, you can use a convenient technique. Put those programs in the \*SCHEMELIB library (the library found in TSCHMELIB data area) and code calling method programs to retrieve the library of the called program from there. For an example of this technique, review the source code for these UTMENU method programs:

UTCHGOBJ, UTCHGOWN, UTCHKOBJ, UTGRTAUT, UTMOVDUP, UTRTVOBJD

## File overrides and opens from CM method programs

Very often, you will use a CM scheme to manage "objects" that are actually represented by records in a database file or files. (UTMENU is an example of such a scheme.) This means that the scheme's method programs typically need to read or update records in these files (to move objects, delete objects, check for objects, and so forth). Because the method programs do not have control over what the library list will be at the time that they are executed, they must execute overrides or user-controlled opens, or both, to these files as needed to ensure that the correct files are opened and processed by the method program. The UTMENU scheme uses two different methods for doing this:

- 1 For examples of method programs that use CL commands to perform the overrides prior to calling RPGLE programs which perform the actual I/O, review the source code for these UTMENU method programs:

UTCHGOWN (calls UTCHGOWNR)  
UTDLTOBJ (calls UTDLTBJR)  
UTRTVOBJD (calls UTRTVOBJDR)  
UTMOVDUP (calls UTMOVDUPR)

- 2 For examples of method programs (actually, programs called by method programs) that perform the overrides in the RPGLE program itself, review the source code for any of these UTMENU method programs:

UTCHGOBJR (called by UTCHGOBJ)  
UTCHKOBJR (called by UTCHKOBJ)  
UTGRTAUTR (called by UTGRTAUT)

---

## Example 2: UTBIO scheme (Employee biographies)

The UTBIO CM scheme manages changes to employee biographies that are stored in an employee biography file. This file and its fields are:

### UTBIO: Employee biography file

| Field           | Description |
|-----------------|-------------|
| Department code | Key         |
| Employee ID     | Key         |
| Biography text  |             |

Because the total length of the file's key is 20 (Department Code and Employee ID are each 10), you cannot identify an employee biography object using a 10-character object name such as the TURNOVER® Lifecycle Manager database supports. Consequently, this scheme uses aliases, which are stored in an alias file. (For more information, see ['Using object aliases in CM schemes' on page 123.](#))

### UTBIOALIAS: Employee biography alias file

| Field              | Description |
|--------------------|-------------|
| Alias              | Key         |
| Department code    |             |
| Option description |             |
| Employee ID        |             |

### UTBIOALILF: Employee biography alias logical file

For retrieving an existing alias for an employee biography, there is a logical file over the alias file.:

| Field           | Description |
|-----------------|-------------|
| Department code | Key         |
| Employee ID     | Key         |

### Restoring the UTBIO scheme

- 1 Sign on as a user profile with authority to restore libraries and TURNOVER® Lifecycle Manager application definitions to the system.
- 2 On a TURNOVER® Lifecycle Manager command line, type `TWRKCMSCH`, and then press Enter.  
The Work with Change Management Schemes panel appears.  
If the UTBIO scheme is already listed, go to step 6.
- 3 Press F8 (Restore).  
The Restore CM Scheme (TRSTCMSCH) panel appears.

- 4 Type `UTBIOSAVF` for the save file name. Accept the default values for the other parameters (the save file is in the TURNOVER® Lifecycle Manager product library). Press Enter.

TURNOVER® Lifecycle Manager restores these items:

- the UTBIO scheme definition
- the UTBIOPRD resource library, which contains the employee biography and alias files (see above), and the UTBIO CM method programs.

- 5 On the main menu, select option 1 (Work with Application Definitions).

If application UT is already on the Work with Application Definitions panel (you might have to page down to see it), go to step 8.

- 6 On a TURNOVER® Lifecycle Manager command line, type `TRSTAPP`.

The Restore Application (TRSTAPP) panel appears.

- 7 For Save file, type `UTAPPSAVF`.

- 8 For Save file library, type the name of your TURNOVER® Lifecycle Manager product library (normally `SOFTTURN`).

- 9 Accept the default values for the other parameters and press Enter.

TURNOVER® Lifecycle Manager restores the UT application definition as well as the UTBIO global type code definition.

- 10 Select TURNOVER® Lifecycle Manager Main Menu option 8 (Utility Menu) and then option 4 (Work with type code definitions).

The Work with Type Codes panel appears.

- 11 Type `2` next to the UTBIO type code, and then press Enter.

- 12 Make sure that the type code definition you now see matches this:

| MM/DD/YY                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |           | Change a Type Code |                  | Your Company, Inc. |       |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|--------------------|------------------|--------------------|-------|
| 16:05:04                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |           |                    |                  | YOURSYS            |       |
| Type code                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | . . . . . | UTBIO              | CM scheme        | . . . . .          | UTBIO |
| Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | . . . . . | Employee Biography | Attribute        | . . . . .          |       |
| Object type                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | . . . . . | *UTBIO             | Source file name | . . . . .          |       |
| Sequence                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | . . . . . | 100                | Data object      | . . . . .          | N     |
| Create command                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | . . . . . |                    |                  |                    |       |
| <div> <div> "&amp;OB" = Object name<br/> "&amp;LI" = Library name<br/> "&amp;TY" = Type code<br/> "&amp;SM" = Source member name<br/> "&amp;SF" = Source file name<br/> "&amp;SL" = Source library name<br/> "&amp;U0" = Generation options<br/> "&amp;U2" = Optimization level<br/> "&amp;U4" = Source listing options<br/> "&amp;U6" = Unassigned<br/> "&amp;U8" = Unassigned<br/> F3=Exit F4=Prompt/select F7=Applications F12=Cancel F22=Long command </div> <div> "&amp;RF" = Reference<br/> "&amp;TO" = Test object name<br/> "&amp;TL" = Test object library name<br/> "&amp;TR" = Target release<br/> "&amp;FM" = From source member name<br/> "&amp;FF" = From source file name<br/> "&amp;FL" = From source library name<br/> "&amp;U1" = Debugging views<br/> "&amp;U3" = Compile options<br/> "&amp;U5" = Unassigned<br/> "&amp;U7" = Unassigned<br/> "&amp;U9" = Unassigned </div> <div> "&amp;X0"-"&amp;X9" = Exit </div> </div> |           |                    |                  |                    |       |

13 Return to the Work with Application Definitions panel, and then type option 14 (Create missing items) next to the UT application.

Unless it has been created previously, you are prompted to create library UTBIODEV (development library). Press Enter to create the library.

14 Now you can create a Programmer Worklist and start experimenting with the UTBIO CM scheme. To start with, you can work with two existing employee biographies:

- Department MARKETING, employee COUNTERB
- Department SALES, employee KRAAPS.

## UTBIO scheme objects

In addition to the files restored during the restore operation (described previously), these objects are restored when you restore the UTBIO CM scheme:

| Object               | Description                                                                                                                                           |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| UTBIO (CLLE)         | The method program for all the scheme's overridden CM methods. These include *CHKOBJ, *CRTDUPOBJ, *DLTOBJ, *MOV OBJ, and *RTVOBJD.                    |
| UTBIOR (RPGLE)       | Called by UTBIO to perform the I/O on the employee biography and alias files.                                                                         |
| UTSLTBIO (CMD, CLLE) | The Select Employee Biography command and command processing program. See <i>'Creating a selection program for user-defined objects' on page 124.</i> |

| Object                 | Description                                                                                                                                                                                                                                              |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UTBIOSLT (DSPF, RPGLE) | The display file and program (called by UTSLTBIO) used to add employee biographies to the Programmer Worklist from a selection list, generating aliases when necessary. See <i>'Creating a selection program for user-defined objects' on page 124</i> . |

The source code for all the UTBIO objects is stored in these source files in the TURNOVER® Lifecycle Manager product library:

- ADDSSRC (PFs, LF, DSPF)
- ACMDSRC (CMD)
- ACLSRC (CLLE)
- ARPGLESRC (RPGLE).

Review these source members, paying special attention to the remaining topics for this UTBIO CM scheme.

## Overriding CM methods to \*NONE

The UTBIO CM scheme uses \*NONE as its the default method program.

\*NONE is appropriate for most of the methods for this particular scheme because they require no action by TURNOVER® Lifecycle Manager but we still want some reflection of a “positive” completion. The exceptions to this are the \*CHKOBJ, \*CRTDUPOBJ, \*DLTOBJ, \*MOV OBJ, and \*RTVOBJD methods, which use UTBIO as the method program; and the \*CHKSRC method, which uses \*TURNOVER.

Using \*NONE for the \*CHKSRC method would cause Programmer Worklists to appear as though source exists at all levels, even though the UTBIO scheme is used to manage a non-source object type (that is, no “&SM”, “&SF”, “&SL” variables in global create command, which is blank). By specifying \*TURNOVER instead, the Programmer Worklists show N in the **Src** column (because TURNOVER® Lifecycle Manager’s method of checking for source is to return N for non-source object types).

Whether you should use \*TURNOVER or \*NONE for a scheme’s method depends on the nature of the object type you are managing and the method you are overriding. When creating your own CM schemes, you might need to experiment some to determine which of these methods is appropriate for which methods (and which methods need to be overridden to programs).

## Using object aliases in CM schemes

TURNOVER® Lifecycle Manager does not support object names longer than 10 characters, so you might need to devise an aliasing scheme to manage user-defined object types that require more than 10 characters for unique identification. The UTBIO CM scheme is an example of this. File UTBIO (Employee Biography File) is keyed by Department Code (length is 10; data type is Character) and Employee (length is 10; data type is Character), which is a total key length of 20

characters. This scheme uses an alias file (UTBIOALIAS) to cross-reference 10-character aliases with the actual department codes and employee IDs that the aliases represent. Program UTBIO (and UTBIOR, which is called by UTBIO), the method program for the \*CHKOBJ, \*CRTDUPOBJ, \*DLTOBJ, \*MOV OBJ, and \*RTVOBJD methods, is responsible for translating aliases to department codes and employee IDs to perform the requested CM method.

In file UTBIO, the format of the Employee ID field is last name, first initial (Bill Smith = SMITHB) and each department (MARKETING, SALES) starts with a unique letter. A logical aliasing scheme, which this sample uses, is to generate aliases in this format:

First letter of department + First letter of Employee ID + Last letter of Employee ID + *nnn*

where *nnn* is 001, 002 (if there is already an alias with the same first three characters), 003, and so on. For example, the alias for Bill Smith in Marketing would be MSB001 (if there is not already an alias starting with MSB).

This aliasing scheme is unique to this example. When creating your own CM schemes, you must devise your own aliasing schemes, using whatever formula is appropriate for your object type.

The UTBIO scheme requires an alias file (UTBIOALIAS) and logical file (UTBIOALILF) to keep track of aliases. However, in some cases it might be possible to use an aliasing scheme that does not require such a file.

For example, suppose you were creating a CM scheme that manages records in a file that is keyed by Category (length is 10; data type is Character) and Part Code (length is 8; data type is Character). The total length of the key exceeds TURNOVER® Lifecycle Manager's limit of 10 for object names. But, suppose you can safely assume that no two categories will begin with the same letter. If that were the case, you could use a scheme where the object alias has the format *Cpppppppp* (where C is the first letter of the Category, and *pppppppp* is the entire part code). Because the alias is not longer than 10 characters, no special alias file is needed.

## Creating a selection program for user-defined objects

As long as the CM scheme is defined appropriately, user-defined object types are fully supported by the Programmer Worklist. However, for user-defined object types, you cannot press F20 and select (using the TURNOVER® Lifecycle Manager user-defined PDM option) objects to add to the worklist from a list of objects or members. You can press F6 and type the object names and type codes to manually add the objects to the worklist. But this could quickly become tedious, and, if your scheme requires an alias file, you also need to add the necessary records to that file before you can add the object to the worklist (using an alias name).

To simplify this, you can write a command that brings up a selection list of your objects and adds selected objects to the active worklist, generating aliases as necessary. This program could then be called from the worklist command line (or by a user-defined worklist option if there is already at least one object on the worklist against which to execute the option).

Such a program would be complex. However, the UTBIO scheme includes such a command (UTSLTBIO) that you can use as is or modify to suit your needs when creating similar commands for your own CM schemes.



Review the source code for these objects:

UTSLTBIO (CMD, CLLE)

UTBIOSLT (DSPF, RPGLE)

### Example 3: UTDDG scheme (JD Edwards data dictionary glossary text)

The **UTDDG** CM scheme manages changes to JD Edwards data dictionary glossary text (referred to in this section as *data dictionary glossary object*). The main driver file is the JD Edwards F9203 data item Alpha descriptions file. The file and its fields are:

#### F9203: Data item alpha descriptions file

| Field  | Description            | Length | Data type |
|--------|------------------------|--------|-----------|
| FRDTAI | Data Item (Key)        | 10     | Character |
| FRLNGP | Language Group (Key)   | 2      | Character |
| FRSYR  | Reporting System (Key) | 4      | Character |
| FRDSCA | Description Alpha      | 40     | Character |
| FRDSCC | Description Compressed | 40     | Character |
| FRSCRN | Screen (Key)           | 10     | Character |

Because the total length of the file's key is 26 (see lengths and data types above for Data Item, Language Group, Reporting System, and Screen), you cannot identify a Data Dictionary Glossary object using a 10-character object name such as the TURNOVER® Lifecycle Manager database supports. Consequently, this scheme uses aliases, which are stored in an alias file. (For additional details, see *'Using object aliases in CM schemes' on page 129.*)

#### UTDDGALIAS: Data dictionary glossary alias file

| Field | Description | Length | Data type |
|-------|-------------|--------|-----------|
| ALIAS | Alias (Key) | 10     | Character |
| DTAI  | Data Item   | 10     | Character |
| LNGP  | Language    | 2      | Character |
| SYR   | System Code | 4      | Character |
| SCRN  | Screen      | 10     | Character |

For retrieving an existing alias for a data dictionary glossary object, there is also a logical file, UTDDGALILF, over the UTDDGALIAS file. The key fields for the UTDDGALILF file are Data Item, Language, System Code, and Screen.

## Restoring the UTDDG scheme

Restoring the UTDDG scheme is slightly different from restoring the UTBIO or UTMENU schemes. Restoring the UTDDG scheme restores the UTDDG library with the method programs, and also creates a TURNOVER® Lifecycle Manager CM scheme. The CM scheme is not assigned automatically to any existing TURNOVER® Lifecycle Manager applications, or to the UT sample application.

### To restore the UTDDG scheme

- 1 Sign on as a user profile with authority to restore libraries and TURNOVER® Lifecycle Manager application definitions to the system.
- 2 On a TURNOVER® Lifecycle Manager command line, type `TWRKCMSCH`, and then press Enter.

The Work with Change Management Schemes panel appears.

If the UTDDG scheme is already listed on this panel, go to step 6.

- 3 Press F8 (Restore).

The Restore CM Scheme (TRSRCMSCH) panel appears.

- 4 Type `UTDDGSAVE` for the save file name. Accept the default values for the other parameters (the save file is in the TURNOVER® Lifecycle Manager product library). Press Enter.  
TURNOVER® Lifecycle Manager restores these items:

- the UTDDG scheme definition
- library UTDDG (the resource library), which contains the data dictionary glossary alias file (see *'UTDDGALIAS: Data dictionary glossary alias file' on page 125*), the UTDDG CM method programs, and the source to the objects being restored.

- 5 On the TURNOVER® Lifecycle Manager main menu, select option 8 (Utility menu,) and then option 4 (Work with type code definitions).

If the UTDDG type code is not on the Work with Type Codes panel (you might have to page down to see it), you need to create it.

If the type code is there, type `2` and press Enter to view it.

Make sure that the type code definition matches this:

| Change a Type Code                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                          | Your Company, Inc.<br>YOURSYS |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|-------------------------------|
| MM/DD/YY<br>16:05:04                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                          |                               |
| Type code . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | UTDDG                    | CM scheme . . . . . UTDDG     |
| Description . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Data Dictionary glossary | Attribute . . . . .           |
| Object type . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | *UTDDG                   | Source file name . . . . .    |
| Sequence . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 100                      | Data object . . . . . N       |
| Create command . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                          |                               |
| <hr/> <div> <div> "&amp;OB" = Object name<br/> "&amp;LI" = Library name<br/> "&amp;TY" = Type code<br/> "&amp;SM" = Source member name<br/> "&amp;SF" = Source file name<br/> "&amp;SL" = Source library name<br/> "&amp;U0" = Generation options<br/> "&amp;U2" = Optimization level<br/> "&amp;U4" = Source listing options<br/> "&amp;U6" = Unassigned<br/> "&amp;U8" = Binding directory<br/> F3=Exit F4=Prompt/select F7=Applications F12=Cancel F22=Long command </div> <div> "&amp;RF" = Reference<br/> "&amp;TO" = Test object name<br/> "&amp;TL" = Test object library name<br/> "&amp;TR" = Target release<br/> "&amp;FM" = From source member name<br/> "&amp;FF" = From source file name<br/> "&amp;FL" = From source library name<br/> "&amp;U1" = Debugging views<br/> "&amp;U3" = Compiler options<br/> "&amp;U5" = Unassigned<br/> "&amp;U7" = Service program source file<br/> "&amp;U9" = Module source file </div> <div> "&amp;X0"-"&amp;X9" = Exit </div> </div> |                          |                               |

- Assign the UTDDG type code to a TURNOVER® Lifecycle Manager application that is used to manage JD Edwards objects. On the main menu, select option 1 (Work with Application Definitions).
- On the Work with Application Definitions panel, type 2 next to a valid JD Edwards application, and then type option 14 (Type codes) next to any level.
- On the Type Codes panel, press F4 (Add/Delete types), page down to the UTDDG type code, type 1 next to it, and then press Enter. When you are prompted to do so, be sure to include this type code for any additional levels of the application.

Now you can create a Programmer Worklist and start experimenting with the UTDDG CM scheme.

## UTDDG scheme objects

In addition to the files restored during the restore operation (described previously), these objects are restored when you restore the UTDDG CM scheme:

| Object         | Description                                                                                                                                                                                     |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UTDDG (CLLE)   | The method program for all the scheme's overridden CM methods. These include *BROWSE, *CHKOBJ, *CRTDUPOBJ, *DLTOBJ, *EDTSRC, *MOV OBJ, and *RTVOBJD.                                            |
| UTDDGR (RPGLE) | Called by UTDDG to perform the I/O on the Data Dictionary Glossary alias file, and to call the JD Edwards routines to transfer Data Dictionary control records (P996301).                       |
| UTDDGC1 (CLLE) | Uses TURNOVER® Lifecycle Manager routines to check if the base Data Dictionary object exists in a library. The base object must exist before a glossary variant can be created in that library. |

| Object                      | Description                                                                                                                                                                                                                                                                     |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UTDDGSLT (CMD, CLLE, DSPF)  | The Select Data Dictionary Glossary object command and command processing program. (See <a href="#">‘Creating a selection program for user-defined objects’ on page 129.</a> )                                                                                                  |
| UTDDGSLTR (RPGLE)           | The display file and program (called by UTSLTDDG) used to add data dictionary glossary objects to the Programmer Worklist from a selection list, generating aliases when necessary. (See <a href="#">‘Creating a selection program for user-defined objects’ on page 129.</a> ) |
| UTDDGEDT (CMD, CLLE, DSPF)  | Sets up the environment to edit a specified Data Dictionary Glossary object, then calls the JD Edwards program P92001 to present the edit screen. When editing a specific screen, you must first position to that screen before starting to edit.                               |
| UTDDGVIEW (CMD, CLLE, DSPF) | Can be specified in a user-defined option to be executed next to a particular worklist item. This presents a window that translates the alias name to the actual Data Dictionary Glossary object.                                                                               |
| UTDDGVIEWR (RPGLE)          | Called by UTDDGVIEW.                                                                                                                                                                                                                                                            |

The source code for all the UTDDG objects is stored in these source files in the resource library UTDDG:

- QDDSSRC (PFs, LF, DSPF)
- QCMDSRC (CMD)
- QCLSRC (CLLE)
- QRPGLSRC (RPGLE).

Review these source members, paying special attention to the remaining topics for this UTDDG CM scheme.

To use these programs in your environment, make sure that the UTDDG library is included in your library list either by adding UTDDG to your TURNOVER® Lifecycle Manager environment record, or by using the `ADDLIB` command.

## Overriding CM methods to \*NONE

The UTDDG CM scheme uses `*NONE` as its default method program. Actually, `*NONE` is appropriate for most of the methods for this scheme. The exceptions to this are the `*BROWSE`, `*CHKOBJ`, `*CRTDUPOBJ`, `*DLTOBJ`, `*EDTSRC`, `*MOV OBJ`, and `*RTVOBJD` methods, which have all been overridden to use program UTDDG as the method program; and the `*CHKSRC` method, which has been overridden to use `*TURNOVER`. For more information, see [‘Overriding CM methods to \\*NONE’ on page 123](#). Except for the scheme name (which in that case is `UTBIO`), the text in the second paragraph of that section also relates to this UTDDG scheme.

## Using object aliases in CM schemes

Because the TURNOVER® Lifecycle Manager database does not support object names longer than 10 characters, you might need to devise an aliasing scheme to manage user-defined object types that require more than 10 characters for unique identification. The UTDDG CM scheme is an example of this. File F9203 (Data Item Alpha Descriptions) is keyed by Data Item (length is 10; data type is Character), Language (length is 2; data type is Character), Reporting System (length is 4; data type is Character), and Screen (length is 10; data type is Character), which is a total key length of 26 characters. This scheme uses an alias file (UTDDGALIAS) to cross-reference 10-character aliases with the actual F9203 records that the aliases represent. Program UTDDG (and UTDDGR, which is called by UTDDG), the method program for the \*BROWSE, \*CHKOBJ, \*CRTDUPOBJ, \*DLTOBJ, \*EDTSRC, \*MOV OBJ, and \*RTVOBJD methods, is responsible for translating aliases to the corresponding F9203 records to perform the requested CM method.

In file F9203, most Data Items consist of 4 positions (even though 10 are available), and the system reporting code must be blank if a screen is being defined. A logical aliasing scheme, which this sample uses, is to generate aliases in the format DDDD\_L\_999:

First 4 positions of the Data Item + underscore + first 1 position of the Language code + an underscore + *nnn*

where *nnn* is 001, 002 (if there is already an alias with the same first three characters), 003, and so forth.

For example, the alias for Data Item ACL1 for language French, screen V12010 would be ACL1\_F\_001 (if there is not already an alias starting with ACL1\_F\_). If the Data item was AD, base language, Screen V069951, underscores are used to fill the blanks, and the alias AD\_\_\_\_\_001 would be used (5 underscores in this example).

You might want to use this aliasing scheme, or create your own that incorporates part of the screen ID. Use whatever scheme works best for your setup. If you change the aliasing scheme, be sure to review all the related programs in the UTDDG scheme.

## Creating a selection program for user-defined objects

As long as the CM scheme is defined appropriately, user-defined object types are fully supported by the Programmer Worklist. However, for user-defined object types, you can not pressing F20 and select (using TURNOVER® Lifecycle Manager's user-defined PDM option) objects to add to the worklist from a list of objects or members. You can press F6 and type the object names and type codes to manually add the objects to the worklist. But this could quickly become tedious. If your scheme requires an alias file, you will need to also add the necessary records to that file before you can add the object to the worklist (using an alias name).

To simplify this, you can write a command that brings up a selection list of available Data Dictionary Glossary objects and adds selected objects to the active worklist, generating aliases as necessary. This program could then be called from the worklist command line (or by a user-defined worklist option if there is already at least one object on the worklist against which to execute the option).

The UTDDG scheme includes a command (UTSLTDDG) that you can use as is or modify to suit your needs.

Review the source code for these objects:

- UTDDGSLT (CMD, CLLE, DSPF)
- UTDDGSLTR (RPGLE)

The UTDDGSLT command hides the system reporting code. It does this because it is not valid to define an override for a screen along with a specific system reporting code.

## 7 Using create commands

This section describes how TURNOVER® Lifecycle Manager handles create commands and how you can customize create commands when promoting objects.

There are two ways to promote an object in TURNOVER® Lifecycle Manager:

- You compile it (CSCO – Copy Source, Compile Object)
- You do not compile it (CSMO – Copy Source, Move Object and CSCD – Copy Source, Create Duplicate Object).

This section focuses on the CSCO promotion method.

When you promote an object, TURNOVER® Lifecycle Manager performs two actions:

- 1 Creates the object: see *'Compiling and creating the object' on page 131*.
- 2 Changes the object after compiling it to apply its original attributes (if TURNOVER® Lifecycle Manager deems this necessary): see *'Changing the object after promotion' on page 132*.

For more information, see 'Changing a form line' in the *TURNOVER® Lifecycle Manager User Guide*.

---

### Compiling and creating the object

TURNOVER® Lifecycle Manager creates the object by compiling, moving, or copying it. When you use the CSCO promotion method, TURNOVER® Lifecycle Manager creates the object by compiling it from promoted source.

It first examines its global create command, which is in the type code for the object. If any of the create command parameters are not specified, TURNOVER® Lifecycle Manager uses the command's default parameters. If the developer has overridden any of the parameter settings, TURNOVER® Lifecycle Manager uses the new values.

#### **Restricting developers from overriding parameter settings in the create command**

You can restrict the developer from overriding, in the development library, any parameter settings in the global create command. You do so by customizing the values in these data areas and file:

- TPROTECT data area
- TKEYPRTF file
- TALWKEYOVR data area

By imposing restrictions using these data areas and file, you are only controlling what the developer can override at the development level. Depending on the promotion method you are using and on the Create parameter (**Create Parm**s) settings, these values can still change during the form run process. For more about how you can customize these data areas and file, see 'Line defaults' in the *TURNOVER® Lifecycle Manager User Guide*.

## Handling create command overrides

Suppose developer A overrides the create command and promotes to QA. Then developer B, checks out the object from QA to her development library (using \*PGMR). Developer B is *not* prompted with developer A's overridden create command. In fact, developer A's object must make its way up to production before other developers can see the create command overrides. However, if developer B does not override the create command, when they build the QA form, TURNOVER® Lifecycle Manager includes developer A's custom create command and changes the Create parameter (**Create Parm**s) to C for the form line.

When developer A builds a level 10 form for promoting from development to QA, any object that has a custom create command will have a Creation parameter (**Create Parm**s) of C (unless the TURNOVER® Lifecycle Manager administrator has locked that setting). The developer can press F4 on that form line to view the overridden create command being used.

When developer A's form is copied, the resulting forms also have, on this line, a value of C for **Create Parm**s (unless the TURNOVER® Lifecycle Manager administrator has locked that setting). This ensures that the object is created correctly at every level.

Because of developer A's custom override, TURNOVER® Lifecycle Manager assumes that this is exactly how developer A wants the object to look in production. Changing the value of **Create Parm**s from P to C ensures that no post-compile changes are performed on the object to change it back to the original object that is being replaced.

---

## Changing the object after promotion

### Controlling how TURNOVER® Lifecycle Manager handles Create parameters

At each level of the application, you can control how TURNOVER® Lifecycle Manager handles the Create parameters (**Create Parm**s) for each object.

| Value | Description |
|-------|-------------|
|-------|-------------|

|   |                                                                                            |
|---|--------------------------------------------------------------------------------------------|
| P | Uses an existing object as the "base" for the attributes of the new object being promoted. |
|---|--------------------------------------------------------------------------------------------|



| Value | Description                                                                                                                                                                                                                                                                                                                                                                         |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | This value indicates “Production,” but TURNOVER® Lifecycle Manager is actually using the libraries defined as targets in the current level and all higher levels, including locked levels. TURNOVER® Lifecycle Manager uses the first object it finds as its “base” object. If it finds no object, TURNOVER® Lifecycle Manager uses a reference object (if you have specified one). |
| T     | TURNOVER® Lifecycle Manager uses “from” object as the base for the attributes of the new object being promoted. The new object replaces the target object’s “broken” parameters.                                                                                                                                                                                                    |

For CSCO, the recommended value is P and no lock.

If the developer customized the create command, TURNOVER® Lifecycle Manager changes the **Create Parm**s value for that line to C (unless the TURNOVER® Lifecycle Manager administrator has locked that setting).

## Using the IBM change command

If **Create Parm**s is set to P and no custom overrides were specified for this promotion, TURNOVER® Lifecycle Manager creates the object in the target location, and then executes the IBM Change command (for example, CHGPF, CHGDSPF, and so on).

TURNOVER® Lifecycle Manager can use only the values of parameters that are available in the Change command. Therefore, some object attributes can only be specified during the creation step and are not available for a post-compile change.

The Change command, when run, applies the final touch to the object, and settings previously made by the compile process could change. This could happen even if those parameters were pre-set in the global create command and “restricted” with the TALWKEYOVR setting.

## 8 Ownership, profiles and jobs

This section describes the ownership for TURNOVER® Lifecycle Manager objects. It explains the values with which TURNOVER® Lifecycle Manager user profiles and job descriptions are created at installation time, and their authority requirements. It also explains how to set up the autostart jobs that TURNOVER® Lifecycle Manager uses to manage communicating with, and distributing changes to, other computers.

---

### Object ownership

The TURNOVER and QSECOFR profiles are owners of TURNOVER® Lifecycle Manager product objects in the shipped product.

#### Changing the ownership

If necessary, you can change the ownership of *all* TURNOVER® Lifecycle Manager product objects to a single profile. The profile you use *must* have \*ALLOBJ authority.

To change the ownership, use the TURNOVER® Lifecycle Manager `SETTOOWN` command. The command checks the product library for a data area called `TURNOWNER` to retrieve the name of the user profile you want to designate (if other than \*DEFAULT). You must create the data area and initialize it with the user profile value you want to use, before you run the command. For more information, contact UNICOM Systems, Inc. Technical Support.

#### Disadvantages of changing TURNOVER® Lifecycle Manager object ownership

- *Additional upgrade steps.* Every time you upgrade TURNOVER® Lifecycle Manager, you must reset the ownership of the product objects by re-running the `SETTOOWN` command.
- *Increased security risk.* The profile you use must have \*ALLOBJ authority to perform the functions necessary on the system. It is unnecessary to create another profile if QSECOFR already exists. The more high-capability profiles you have to protect from unauthorized access, the higher your security risk.

- *Authority issues.* The TURNOVER® Lifecycle Manager authority programs grant user profile TURNOVER \*ALL authority to all of the TURNOVER® Lifecycle Manager objects. Even if you change TURNOVER® Lifecycle Manager object ownership to another profile, you cannot remove the TURNOVER profile to reduce the number of high risk profiles, because TURNOVER needs to be able to update its own files.

---

## User profiles

TURNOVER® Lifecycle Manager uses a set of user profiles to perform different kinds of work. Most of these profiles are created when you install TURNOVER® Lifecycle Manager; the rest you must create when you decide to configure the TURNOVER® Lifecycle Manager features that require them.

### Profiles created during TURNOVER® Lifecycle Manager installation

TURNOVER® Lifecycle Manager's first-time installation process for development computers creates the TURNOVER, TSERVER, TURNESCQ, and TOMSGRCV user profiles.

Values that are not listed in the following user profile attribute lists accept the CRTUSRPRF command keyword defaults.

#### TURNOVER profile

The TURNOVER profile is created with these attributes:

| Attribute | Value                    |
|-----------|--------------------------|
| USRPRF    | TURNOVER                 |
| PASSWORD  | &PASSWORD                |
| USRCLS    | *SYSOPR                  |
| CURLIB    | &PRDLIB                  |
| INLPGM    | &PRDLIB/TOINIT           |
| INLMNU    | *SIGNOFF                 |
| LMTCPB    | *NO                      |
| TEXT      | 'TURNOVER Administrator' |
| SPCAUT    | *JOBCTL *SPLCTL          |
| OUTQ      | &PRDLIB/TURNOVER         |

The TURNOVER profile is used by the TURNOVER® Lifecycle Manager administrator to configure the TURNOVER® Lifecycle Manager product.

Limit access to the TURNOVER user profile. Any user who has access to this profile can delete and change TURNOVER® Lifecycle Manager product objects.

### TOMSGRCV profile

The TOMSGRCV user profile is created with these attributes:

| Attribute | Value                                    |
|-----------|------------------------------------------|
| USRPRF    | TOMSGRCV                                 |
| PASSWORD  | *NONE                                    |
| TEXT      | 'TURNOVER Confirmation Message Receiver' |

TURNOVER® Lifecycle Manager uses this user profile to receive form processing and confirmation messages on the development system, from both local and remote systems.

This profile's password is set to \*NONE, so that users cannot sign onto your system as user TOMSGRCV.

### TURNESCQ profile

The TURNESCQ user profile is created with these attributes:

| Attribute | Value                                       |
|-----------|---------------------------------------------|
| USRPRF    | TURNESCQ                                    |
| PASSWORD  | *NONE                                       |
| TEXT      | 'TURNOVER Project Escalation Queue Profile' |

TURNOVER® Lifecycle Manager uses the TURNESCQ profile on Development copies of TURNOVER® Lifecycle Manager, to perform project escalation and associated messaging.

This profile's password is set to \*NONE, so that users cannot sign onto your system as user TURNESCQ.

### Additional TURNOVER® Lifecycle Manager user profiles

If you use TURNOVER® Lifecycle Manager's Integrated File System (IFS) support or FTP distribution features, you might need to create some additional user profiles for those features to use. In some cases, already-existing profiles can be used.

For more information, see ['Managing IFS objects' on page 197](#)

### IFS adopted authority user

You can set up TURNOVER® Lifecycle Manager so that, when it is promoting files that reside in your IBM i IFS, it adopts the authority of a specific user profile.

You identify this user profile by providing its name in the **IFS Adopted Authority User** field in the TURNOVER® Lifecycle Manager global defaults (main menu option 8, then option 5). The default name is TURNIFS, but this profile is not shipped with TURNOVER® Lifecycle Manager; you must create it.

## TURNIFS profile

Create your TURNIFS user profile with these suggested attributes:

| Attribute | Value                                                                                                                                                                                     |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| USRPRF    | TURNIFS                                                                                                                                                                                   |
| PASSWORD  | *NONE if managing IFS objects only; a password is required for managing QNTC objects or if the TURNOVER® Lifecycle Manager mail daemon also uses the TURNIFS profile. (See INLMNU below.) |
| TEXT      | 'TURNOVER® Lifecycle Manager IFS Adopted Authority User'                                                                                                                                  |
| INLMNU    | *SIGNOFF if you are using features for which this profile requires a password. (See PASSWORD.)                                                                                            |
| SPCAUT    | *ALLOBJ (See Note, below.)                                                                                                                                                                |
| USRCLS    | *SECOFR, if SPCAUT is not *ALLOBJ (See Note, below.)                                                                                                                                      |

**Note** The profile must either have \*ALLOBJ special authority, *or* be a member of the \*SECOFR user class, *or* have appropriate authority for the specific IFS objects that TURNOVER® Lifecycle Manager will manage.

## Recommendation

Without either \*ALLOBJ authority OR \*USE authority to all necessary user profiles, a profile may not be able to manage an IFS object's primary group profile or its owner. Read IBM's knowledge base document number 18314911, "*IFS Authority Considerations*", when you are preparing to configure TURNOVER® Lifecycle Manager for managing objects that are stored in your IFS directories.

---

## QNTC adopted authority user

You can set up TURNOVER® Lifecycle Manager so that, when it is promoting files stored in the /QNTC file system, it adopts the authority of a specific user profile that you create. (The QNTC file system on an IBM i computer allows the operating system to access shared directories that are on computers in a Windows network.)

You identify this user profile by providing its name in the **QNTC Adopted Authority User** field in the TURNOVER® Lifecycle Manager global defaults (main menu option 8, then option 5).

The TURNOVER® Lifecycle Manager Mail Daemon also uses this profile: see *'TURNOVER® Lifecycle Manager mail daemon user profile' on page 139* and *'Setting up the mail daemon' on page 236*.

A Windows domain user with the same name and password must be defined to your Windows network, and this profile must have authority for the Windows share as well as the appropriate directories within.

### Recommendation

Use the same profile for your QNTC Adopted Authority User as you use for your IFS Adopted Authority User, and assign it (and its Windows counterpart) any specific IFS and QNTC share and directory permissions it will need. This is because all TURNOVER® Lifecycle Manager IFS object archiving (regardless of which file system the objects originate from) occurs in the IFS root, requiring the IFS profile. By giving the IFS profile the QNTC-related authorities it needs and specifying it as your QNTC Adopted Authority User, you avoid creating an additional user profile that is essentially identical to another profile that is already in use.

See IBM's knowledge base article number 18314911 to locate additional information about the implications of IFS authority settings.

---

## User profile for FTP

TURNOVER® Lifecycle Manager uses this user profile on the target computer, when performing FTP distributions.

You identify this user profile when overriding a system's network distribution method to use the \*FTP method (Distribution Menu option 6).

The FTP user profile does not require any special authorities. It can be any user profile *on the target computer* that has sign-on capability and sufficient authority on that computer for creating save files, and issuing commands.

These are the recommended settings:

| Attribute | Value               |
|-----------|---------------------|
| USRPRF    | TURNFTP             |
| PASSWORD  | <i>YourValidPWD</i> |
| PWDEXP    | *NO                 |
| STATUS    | *ENABLED            |
| USRCLS    | *SECOFR             |
| ASTLVL    | *SYSVAL             |
| CURLIB    | *CRTDFT             |

| Attribute | Value                                          |
|-----------|------------------------------------------------|
| INLPGM    | *SIGNOFF                                       |
| INLMNU    | *SIGNOFF                                       |
| LMTCPB    | *NO                                            |
| TEXT      | 'TURNOVER® Lifecycle Manager FTP Distribution' |
| GRPPRF    | *NONE                                          |
| OWNER     | *USRPRF                                        |

### Recommendations

- Set the INLMNU parameter to \*SIGNOFF for the initial program, and combine that setting with a password that never expires. This ensures that distribution will continue to work uninterrupted (no-one must intervene to reset an expired password), and prevents users from accessing the system using this profile.
- Set LMTCPB to \*NO. This profile must be able to issue commands.

---

## TURNOVER® Lifecycle Manager mail daemon user profile

When you install and configure the TURNOVER® Lifecycle Manager mail daemon (see [‘Setting up the mail daemon’ on page 236](#)) you must specify a user profile value in the QNTC Adopted Authority User global default.

The profile you use for the mail daemon does not require any special authorities. It must have a valid password and it cannot be \*DISABLED. The TURNOVER® Lifecycle Manager mail daemon uses the value you specify in the QNTC Adopted Authority User profile for this purpose. (You should use the TURNIFS profile for both the QNTC and the IFS adopted authority settings.)

---

## Autostart jobs

TURNOVER® Lifecycle Manager uses an operating system feature called *autostart jobs* to handle work that needs to be completed upon request. These optional TURNOVER® Lifecycle Manager features that, when configured as recommended, handle work using autostart jobs:

| Feature                                             | Description                                                                                                                                                                                                                                                                                                          | Autostart job |
|-----------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| Distribution confirmation messaging                 | When you want a development computer to receive confirmation messages from a production computer after changes have been distributed, you can configure automatic confirmation messaging on your development computer.                                                                                               | TOMSGRCV      |
| Automatic distribution receive/install (SNADS only) | When you want a production computer to automatically receive and install changes that are arriving from a development computer, you can configure automatic distribution on your remote computer.                                                                                                                    | TURCRARE      |
| Task escalation                                     | When you want your TURNOVER® Lifecycle Manager project system to automatically “escalate” tasks to a higher priority if response goals are not met and notify people when there is work to be done, you can configure TURNOVER® Lifecycle Manager’s automatic task escalation features on your development computer. | TURNESCQ      |
| Dynamic Wisedesk Updating                           | When you want your Wisedesk knowledge base trees to be automatically updated when new tasks enter the system, you can configure TURNOVER® Lifecycle Manager’s Wisedesk Dynamic Update feature on your development computer.                                                                                          | WISEDESK      |

## Setting up the autostart jobs

To set up these autostart jobs, you add an autostart job entry for each desired job to an appropriate subsystem description. In order for a feature to begin processing its data requests, the associated job must be running in the subsystem where it has been added. These jobs can be set up either automatically using TURNOVER® Lifecycle Manager’s autostart job setup utility, or manually by creating the necessary autostart job entries.

### Recommendation

Use the automatic autostart job setup procedure whenever possible. It runs automatically during a first-time install; to run it again, use the TURNOVER® Lifecycle Manager Setup menu.



## Automatically setting up autostart jobs

The TOMSGRCV, TURNESCQ, and WISEDESK autostart jobs run on your development computers, and are automatically set up and started for you in subsystem TSERVER during the installation procedure.

The TURCRARE job is needed only for SNADS distribution and runs only on remote computers. It is not set up automatically during the installation process, unless someone changes the default settings at that time. However, you can revisit the automatic setup utility to set up that job using the instructions provided here.

You can also automatically associate the autostart jobs with a subsystem that you specify (the TSERVER subsystem is the default).

### To set up the autostart jobs

- 1 In TURNOVER® Lifecycle Manager, access the command line, and then type:

```
GO TOSETUP
```

The TURNOVER® Lifecycle Manager Setup Menu appears.

TOSETUP
**TURNOVER® Lifecycle Manager Setup Menu**
System: YOURSYS

Select one of the following:

- 1. Configure APPC Communications
- 2. Configure TCP/IP Communications
- 3. Configure Autostart Jobs

**Bottom**

Selection or command  
 ==> \_\_\_\_\_

---

F3=Exit
F4=Prompt
F6=Display messages
F9=Retrieve
F12=Cancel

F13=Information assistant
F14=Work with submitted jobs
F24=More keys

(c) Copyright 2005 - 2018 UNICOM Systems, Inc.

- 2 Select option 3 (Configure Autostart Jobs) to set up and start the autostart jobs.

The Add TURNOVER® Lifecycle Manager Autostart Jobs panel appears.

**Add TURNOVER® Lifecycle Manager Autostart Jobs (TCSADDAJE)**

Type choices, press Enter.

|                                  |         |             |
|----------------------------------|---------|-------------|
| Subsystem name . . . . .         | TSERVER | Name        |
| Library . . . . .                | QGPL    | Name, *LIBL |
| Add Wisedesk Dynamic Update . .  | *YES    | *YES, *NO   |
| Add Escalation Queue Monitor . . | *YES    | *YES, *NO   |
| Add Form Message Receiver . . .  | *YES    | *YES, *NO   |
| Add Form Auto Receiver . . . . . | *NO     | *YES, *NO   |

**Bottom**

F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display  
 F24=More keys

- 3 Specify the name of the subsystem, and its library, where you want TURNOVER® Lifecycle Manager to set up the autostart jobs. (The TSERVER subsystem is the default.)

**Add Form Auto Receiver** controls setup of the TURCRARE job. For more information, see [‘Setting up the TURCRARE job on a remote system’ on page 142.](#)

- 4 Press Enter.

TURNOVER® Lifecycle Manager sets up the autostart jobs you requested.

- 5 End and restart the subsystem (after ensuring that no work is in progress) to start the jobs. Any time the subsystem is ended and restarted, these jobs also end and restart.

## Setting up the TURCRARE job on a remote system

- 1 Sign onto the remote computer, and then access TURNOVER® Lifecycle Manager.
- 2 On Setup menu, run option 3.
- 3 On the Add TURNOVER® Lifecycle Manager Autostart Jobs panel, type \*NO for the Wisedesk, Escalation, and Message Receiver jobs. Type \*YES for the Form Auto Receiver job.
- 4 Press Enter.

TURNOVER® Lifecycle Manager sets up the Form Auto Receiver job in the specified subsystem. To start the job, you must end and restart the subsystem (ensure that no work is in progress). Remember that any time the subsystem is ended and restarted, the Form Auto Receiver job also ends and restarts.

## Manually setting up autostart jobs

### Setting up the confirmation message monitor job (TOMSGRCV)

This job runs on the development system, that is, the system from which you distribute changes. If you are not distributing forms to other computers, you do not need this job.

Follow these instructions to manually set up the confirmation message monitor job TOMSGRCV. The message receiver job must be active for TURNOVER® Lifecycle Manager to receive and process confirmation messages that production computers send back to the development computer after a distribution completes.

To set up the confirmation message receiver job, you can either:

- set up an autostart job that is started when subsystem TSERVER (or another subsystem you choose) is started
- submit a job when you IPL your system.

### Setting up TOMSGRCV as an autostart job

Type the following one-time ADDAJE (Add Autostart Job Entry) command:

```
ADDAJE SBSD(TSERVER) JOB(TOMSGRCV) JOBD(SOFTTURN/TOMSGRCV)
```

You can name the job anything you want in your ADDAJE command. However, you must use the exact job description (JOBD) that is shown here.

#### Notes

- Substitute an appropriate subsystem name for TSERVER, if you prefer to use a different subsystem.
- Substitute your product library name for SOFTTURN, if different.
- If you specify a subsystem other than the TSERVER default, then you must adjust the **Job queue** parameter of the TOMSGRCV job description in your TURNOVER® Lifecycle Manager product library to identify the job queue that is associated with the subsystem you chose.

When you install TURNOVER® Lifecycle Manager, the install or upgrade procedure creates a user profile called TOMSGRCV and adds a directory entry for it on your system. This is the user profile to which confirmation messages are sent after a distributed form has run on a target computer (if you set the system distribution defaults to do this).

- 1 If you are running one or more additional versions of this job to service other TURNOVER® Lifecycle Manager environments, you must create a corresponding user profile where the confirmation messages for that environment should be sent. You must also create a separate TOMSGRCV job description for each environment. To do this, just copy and adjust the TOMSGRCV profile and job description from your live environment.

- 2 When a confirmation message is received in the user profile's message queue, the monitor job is called and processes the confirmation, updating the distribution history for the TURNOVER® Lifecycle Manager form. For more information, see 'Distributing Changes to Production Computers' in the *TURNOVER® Lifecycle Manager User Guide*.

## Submitting TOMSGRCV as a standard job

(This is recommended only as a short-term solution.)

To submit the confirmation message monitor job without bringing down the subsystem, you can type this command on a command line, or include it in your startup CL program, for all instances of the job that you plan to run:

```
SBMJOB JOB (TOMSGRCV) JOBD (SOFTTURN/TOMSGRCV) JOBQ (QINTER) USER (TOMSGRCV)
RQSDTA (*JOB) INLLIBL (*JOB)
```

### Notes

- If you submit this as a standard job, it will not end and restart automatically when the subsystem is ended and restarted.
- Substitute your product library for SOFTTURN, if different.
- If you submit this as a standard job, make sure that the job queue to which you are submitting it allows a sufficient number of active jobs. Use the `DSPSBSD` (Display Subsystem Description) command and select **Job queue entries** to view the **Max Jobs** parameter.

---

## Setting up the auto-receive monitor job (TURCRARE)

This job runs on your production systems where you receive changes that were distributed from a development computer via SNADS.

If you are using FTP instead of SNADS to distribute forms to other computers, you can bypass this procedure.

This job is not set up for you automatically during a first-time installation procedure, because the install procedure assumes that you are installing on a development computer. Follow these instructions to set up the auto receive monitor job `TURCRARE`. The auto receive job must be active for TURNOVER® Lifecycle Manager to automatically receive and automatically process forms sent from development computers to one or more production computers.

To set up the auto receive monitor job, you can either:

- set up an autostart job that is started when subsystem TSERVER (or another subsystem you choose) is started, OR
- submit a job when you IPL your system.

In either case, this job waits for a distribution message to appear in the target user's message queue (user TURNOVER by default), and then receives the distribution.

## Setting up the TURCRARE job as an autostart job

- 1 Sign on to the receiving system.
- 2 Type the following one-time ADDAJE (Add Autostart Job Entry) command:

```
ADDAJE SBSDD(TSERVER) JOB(TURCRARE) JOBD(SOFTTURN/TURCRARE)
```

You can name the job anything you want in your ADDAJE command. However, you must use the exact job description (JOBD) that is shown here.

### Notes

- The TURCRARE job name is not necessarily the name you will use. The names you give your jobs will depend on your own particular needs.
- Substitute the appropriate subsystem name for TSERVER, if different. Change the **Job queue** parameter of the TURCRARE job description in library SOFTTURN to identify the job queue associated with the subsystem you choose.
- Substitute your product library for SOFTTURN, if different.
- If this job is running, make sure the **Receive forms method** field is set to \*AUTOSTART and the **Automatically submit received forms** field is set to Y in the remote computer's system Receive distribution defaults. (On the remote production computer, select Main Menu option 11, then option 1; then select the production machine's system definition entry with option 9.)
- Note that the System Receive Distribution Default settings on a production system can be overridden by the sending system – either at the system or the application level, depending on whether the production system is set up to accept overrides. Check the **System Receive Distribution Defaults** settings on the production computer to avoid unexpected results. (The sending system can override at the system OR application level if the **Allow override from sending system** field is set to Y in the production computer's **Receive Distribution Defaults**.)

## Submitting TURCRARE as a standard job

(This is recommended only as a short-term solution.)

To submit the TURCRARE job without bringing down the subsystem, you can type this command or include it in your startup CL program:

```
SBMJOB JOB(TURCRARE) JOBD(SOFTTURN/TURCRARE) JOBQ(QINTER) USER(TURNOVER)  
INLLIBL(*JOBDD) HOLD(*NO) RQSDTA(*JOBDD)
```

### Notes

- Realize that if you submit this job as a standard job, it does not restart when the subsystem is ended and restarted.
- Substitute your product library for SOFTTURN, if different.

- If you submit this as a standard job, make sure that the job queue allows a sufficient number of active jobs. Use the `DSPBSBD` (Display Subsystem Description) command and select **Job queue entries** to view the **Max Jobs** parameter.
- The user profile you supply for the `USER` parameter must be the **Target User** for this system as it appears in the system distribution default entry on your development computer (see the **System Distribution Defaults** and the **Target User** parameter).
- This command must be run when the target user is not signed on, because this job must be able to allocate the target user's message queue. If the target user signs on frequently and needs to have the message queue in break mode, consider using another user profile for the target user.
- The auto receive process submits the `RCVTONETF` job to the job queue identified by the `RCVJOBQ` data area. In turn, the `RCVTONETF` job submits the `TOAUTORCV` job to the job queue identified in the **TURNOVER® Lifecycle Manager** job description. If you want to change the job queue to which `TURCRARE` submits its network file receive work, change `DTAARA RCVJOBQ` in library `SOFTTURN` (or your **TURNOVER® Lifecycle Manager** data library). This data area is 20 characters long: ten characters for the job queue name, and ten for job queue library name.
- Use the `WRKUSRJOB USER(targetuserid)` command to see the jobs that `TURCRARE` submits when processing.

---

## Setting up the escalation queue monitor job (TURNESCQ)

**Note** Before following these instructions, make sure such a job was not already created during the installation procedure.

Follow these instructions if the Project system's escalation queue monitor is not running anywhere on your system, and you want to use the **TURNOVER® Lifecycle Manager** project escalation messaging features.

The escalation queue monitor reviews the *escalation queue* at intervals you determine to decide whether or not to send alarm messages or escalate a task's status or priority.

To set up the escalation queue monitor, you can either:

- Set up an autostart job that is started when subsystem `TSERVER` (or another subsystem you choose) is started, OR
- Submit a job when you IPL your system.

In either case, this job wakes up at the designated interval to process the escalation queue, then goes back to sleep.

### Setting up the TURNESCQ job as an autostart job

Type the following one-time `ADDAJE` (Add Autostart Job Entry) command:

```
ADDAJE SBSD(TSERVER) JOB(TURNESCQ) JOBD(SOFTTURN/TURNESCQ)
```

- Substitute the appropriate subsystem name for TSERVER, if different. Be sure to adjust the **Job queue** parameter of the `TURNESCQ` job description in library `SOFTTURN` to identify the job queue associated with the subsystem you choose.
- The `TURNESCQ` job name is not necessarily the name you will use. The names you give your jobs will depend on your own particular needs.
- Substitute your product library for `SOFTTURN`, if different.
- You can name the job anything you want in the `ADDAJE` command. However, you must use the exact job description (JOBID) that is shown here.
- If you are setting up one or more additional versions of the escalation queue monitor job to monitor the escalation queue for other **TURNOVER®** Lifecycle Manager environments, be sure you have created and adjusted a job description and a user profile for the new job to use.

## Submitting TURNESCQ as a standard job

(This recommended only as a short-term solution.)

To submit the `TURNESCQ` job without bringing down the subsystem, you can type this command or include it in your startup CL program:

```
SBMJOB CMD(TSTRESCQ INTERVAL(900))
JOB (TURNESCQ)
JOBID (SOFTTURN/TURNESCQ)
```

- If you submit this job as a standard job, it does not restart when the subsystem is ended and restarted.
- Substitute your product library for `SOFTTURN`, if different.
- If you submit this as a standard job, make sure that the job queue allows a sufficient number of active jobs. Use the `DSPSBSD` (Display Subsystem Description) command and select **Job queue entries** to view the **Max Jobs** parameter.
- Change the interval at any time by rerunning the `SBMJOB` command, specifying a new interval. If the escalation queue monitor is running, only the job's wake-up interval changes.

---

## Setting up the Wisedesk dynamic update monitor job

### Setting up the WISEDESK job as an autostart job

Type the following `ADDAJE` (Add Autostart Job Entry) command:

```
ADDAJE SBSD(TSERVER) JOB (WISEDESK) JOBID (SOFTTURN/WDAUTO)
```

- The `WISEDESK` job name is not necessarily the name you will use. The names you give your jobs depend on your own particular needs.

- Substitute the appropriate subsystem name for TSERVER, if different. Be sure to adjust the **Job queue** parameter of the WDAUTO job description in library SOFTTURN to identify the job queue associated with the subsystem you choose.
- Substitute your product library for SOFTTURN, if different.

### Submitting WISEDESK as a standard job

(This should be used only as a short-term solution.)

To submit the autostart job using the WDAUTO command without bringing down the subsystem, you can type this command (or include it in your CL startup program):

```
WDAUTO DTAQ(queuename) DTAQLIB(libname)
```

where *queue**name* is the name of your data queue and *lib**name* is the name of your data queue library.

- If you submit this job as a standard job, it does not restart when the subsystem is ended and restarted.
- Substitute your product library for SOFTTURN, if different.
- If you submit this as a standard job, make sure that the job queue to which you submit it allows a sufficient number of active jobs. Use the DSPSBSD (Display Subsystem Description) command and select **Job queue entries** to view the **Max Jobs** parameter.



## 9 Security recommendations

These security recommendations ensure that there is a record of every change to production source members and objects in TURNOVER® Lifecycle Manager.

Review the security on all of your production libraries and objects to ensure that changes can only be made through approved channels.

Although you may have authority to some functions of TURNOVER® Lifecycle Manager such as to checkout source or to submit a form, you need no special authorities outside of TURNOVER® Lifecycle Manager. As a programmer or manager, you should not be able to access any more than the object description of production objects.

Maintaining security requires continually monitoring your security exposure and understanding where the potential leaks may be.

---

### Object owners

- Every object on your system has a designated owner.
- Create a profile that owns all objects in an application and give it a password of \*NONE. That way, no one can sign on as that user and assume the owner's object rights.
- Give appropriate authority to each user that needs access to the application through authorization lists or on the individual objects themselves.
- Alternatively, create an additional user profile with limited rights to the objects in the applications' libraries. This second profile can be used as a group profile for all who need to access the applications. This protects the integrity of both the data and the objects by not allowing excess privileges to any user or developer.
- In TURNOVER® Lifecycle Manager, objects promoted to each application level can have a different owner. When objects are added or replaced, the owner is changed to the profile name you provide (see 'Working with application definitions' in the *TURNOVER® Lifecycle Manager User Guide*). Object owners can also vary by object type by assigning ownership using reference objects.

- Do not use QSECOFR as the owner of any of your applications. The primary reason for not using QSECOFR is the ability to adopt authority in a program from the object owner. If the program in question presents a command line to the user, they will have all of the authority of QSECOFR.
- Limit the authority of your programmers so that they can update objects in production without using of TURNOVER® Lifecycle Manager. Otherwise, you have no control except to the extent that programmers use TURNOVER® Lifecycle Manager voluntarily. You still benefit from the convenience that TURNOVER® Lifecycle Manager provides, but the primary benefit of security will be undermined. TURNOVER® Lifecycle Manager has the ability to monitor changes not made through TURNOVER® Lifecycle Manager by running an audit report. For more information about this feature, see 'Reports' in the *TURNOVER® Lifecycle Manager User Guide*. TURNOVER® Lifecycle Manager also alerts you that a source member or object has been changed outside of TURNOVER® Lifecycle Manager when you try to check out a source member or object and when you try to replace it during a promotion job.

---

## Production libraries

Use libraries to separate and secure each application. The ability to define and secure libraries enables you to manage objects for each application more efficiently. You can secure each library to limit access to objects stored there.

After you create your application definitions, run the Check Application Definition command (CHKAPPDFN) to uncover any objects that do not match the object ownership defined for your application. Do this regularly and make sure your technical staff is aware of it.

The DSPOBJAUT (Display Object Authority) panel might look like this for the library PAYROLL:

| User    | Object    | ---Object--- |     |       | -----Data----- |     |        |        |
|---------|-----------|--------------|-----|-------|----------------|-----|--------|--------|
|         | Authority | Opr          | Mgt | Exist | Read           | Add | Update | Delete |
| QSECOFR | *ALL      | X            | X   | X     | X              | X   | X      | X      |
| PAYMGR  | *USR DEF  | X            | X   |       | X              | X   | X      | x      |
| PAYDEPT | *USE      | X            | X   |       | X              |     |        |        |
| *PUBLIC | *EXCLUDE  |              |     |       |                |     |        |        |

The Owner is PAYMGR, who has altered their own authority to the library so that they do not inadvertently delete the library (Exist). They can maintain authority to the library for others, including QSECOFR. They can change their own authority. They have all rights to all objects in the library.

PAYDEPT is a group user. All members of the payroll department share the authority of user PAYDEPT (each as GRPPRF(PAYDEPT) in his/her user profile). (Alternatively, an authorization list or a list of individual user profiles could be used.) Everyone in each group has access to the library, but members of the group can still be restricted from specific objects within the library.

If a particular program within the application is designed to add or delete an object (such as a file member), that program could be changed to adopt the authority of the object owner (in this case PAYMGR) to enable the add or delete operation to take place.

On the IBM i computer, \*PUBLIC is specifically excluded to ensure that, no matter how object authority was set up, only users authorized at the library level could access objects in the library. The user would see a "Not authorized to library" message, and could not get into any object in the library no matter how the object authorities were defined.

When viewing the DSPOBJAUT panel for a library, keep in mind that "Data" means entities within the object displayed; in this case, the library. In other words, "Data" rights are rights to the objects in the library, not the data within those objects. You need to grant authority to objects in the library to grant data rights: see *'Setting specific object authority' on page 151*.

---

## Setting specific object authority

Generally, objects are of two types:

- production files (physical and logical files) and
- program-related objects (programs, message files, display files, and so on).

To set up authority reference objects for use during form processing, see *'Setting up authority reference objects' on page 152*.

### Production files

Specific authority granted to a user, or groups of users, is usually \*CHANGE (IBM i) and includes:

| User     | Object Authority | ----Object----- |     |       |  | -----Data----- |     |        |        |
|----------|------------------|-----------------|-----|-------|--|----------------|-----|--------|--------|
|          |                  | Opr             | Mgt | Exist |  | Read           | Add | Update | Delete |
| OBJUSER  | *CHANGE          | X               |     |       |  | X              | X   | X      | X      |
| OBJOWNER | *ALL             | X               | X   | X     |  | X              | X   | X      | X      |

The object owner has all rights to the object. The object users usually have:

- Operational rights
- All data rights (Read, Add, Update, Delete)

### Programs and related objects

Specific authority granted to a user, or groups of users, is usually \*USE (IBM i) and includes:

| User     | Object Authority | ----Object----- |     |       |  | -----Data----- |     |        |        |
|----------|------------------|-----------------|-----|-------|--|----------------|-----|--------|--------|
|          |                  | Opr             | Mgt | Exist |  | Read           | Add | Update | Delete |
| OBJUSER  | *CHANGE          | X               |     |       |  | X              |     |        |        |
| OBJOWNER | *ALL             | X               | X   | X     |  | X              | X   | X      | X      |

The object owner has all rights to the object. The object users need only:

- Operational rights
- Read rights

## Recommendation

Secure your objects with authority lists instead of using special authority; these are easier to maintain. Do not mix authority lists and special authorities; performance will suffer because of the additional checking done by the operating system.

## Source files and members

Production source files need to be secured as well. There is both library and file authority to consider.

If you keep production source files in a separate library, grant authority as follows:

### Source library

| User     | Object    | ----Object----- |     |       | -----Data----- |     |        |        |
|----------|-----------|-----------------|-----|-------|----------------|-----|--------|--------|
|          | Authority | Opr             | Mgt | Exist | Read           | Add | Update | Delete |
| OBJUSER  | *CHANGE   | X               | x   | X     | X              | X   | X      | X      |
| OBJOWNER | *ALL      | X               |     |       | X              |     |        |        |

The programmer cannot add any files or members to the library. They would need 'Add' rights for the library to be able to add a member.

### Source files

| User     | Object    | ----Object----- |     |       | -----Data----- |     |        |        |
|----------|-----------|-----------------|-----|-------|----------------|-----|--------|--------|
|          | Authority | Opr             | Mgt | Exist | Read           | Add | Update | Delete |
| OBJUSER  | *CHANGE   | X               | x   | X     | X              | X   | X      | X      |
| OBJOWNER | *ALL      | X               |     |       | X              |     |        |        |

Programmers cannot add or update members. They can only view the member list and view the contents of a member. If your programmers are not part of group profile QPGMR, then secure them by authority list or by specifying their user profiles in the object authority list.

Unfortunately, you cannot set file authority to allow programmers to see a list of files and members, but not their contents. If you do not want programmers to be able to view a member list or the contents of a source file other than through TURNOVER® Lifecycle Manager, you could remove the 'Read' rights to the source files. This would still be adequate for checking out source by object, where you would work with an object list in PDM and check out the source for an object using a TURNOVER® Lifecycle Manager user-defined command.

---

## Setting up authority reference objects

Set up dummy objects (REFxxxx where xxxx is PF, LF, RPG, CLP, MSGF, or DSPF) in one or more production libraries. Then, when you define the **Create parameters** and **Authority parameters** in the Line Defaults of the application definition, you can reference these objects by setting the **Reference object** parameters to REF\*. TURNOVER® Lifecycle Manager references the named object when it grants authority to objects you submit to be promoted. They can reside in any library you choose.

If you use `REF*` in the **Reference object** portion of the **Create parameters** and **Authority** fields, TURNOVER® Lifecycle Manager parses `REF` with the object type attribute code, such as `PF` or `CLP` and then looks for a reference object with that name, for example `REFCLP`. When you create reference objects, TURNOVER® Lifecycle Manager maintains object authorities for you. You can copy these objects in the SOFTTURN library to get started:

| Name      | Description             |
|-----------|-------------------------|
| REFPF     | Reference physical file |
| REFLF     | Reference logical file  |
| REFRPG    | Reference program       |
| REFCLP    | Reference program       |
| REFCMD    | Reference command       |
| REFMSGF   | Message file            |
| REFDSPF   | Reference display file  |
| REFPRTF   | Reference print file    |
| REFDTAARA | Reference data area     |

Although TURNOVER® Lifecycle Manager permits you to reference any object on your system, if you use object types other than those on the list, be sure to create your own `REFxxxx` objects so that you can use the `REF*` feature consistently in all of your applications.

---

## Object distribution and security

When TURNOVER® Lifecycle Manager distributes objects, it does not transfer private authorities to the destination computer. The IBM i operating system retains private authorities with user profiles, not with objects. Therefore, in TURNOVER® Lifecycle Manager, you cannot use a **Create parameters** setting of `T` (Test) on a remote computer or preserve private authorities by moving an object to a remote computer.

- If user profiles have private authority to an object that is being restored, those private authorities are usually not affected. (However, restoring certain types of programs can cause private authorities to be revoked.)
- If an object is deleted from the system and then restored from a saved version, private authority for the object no longer exists on the system. When an object is deleted, all private authority to the object is removed from user profiles.
- If you need to recover private authorities, you must use the Restore Authority (`RSTAUT`) command. The normal sequence is:
  - i Restore user profiles

- ii Restore objects
- iii Restore authority.

---

## Checking your system for security exposure

### Check the application definition reports

After you have defined applications to TURNOVER® Lifecycle Manager, run this report (see TURNOVER® Lifecycle Manager Reporting menu) to show you ownership, authority and other potential problems you may have to correct. When you are sure that the application definitions are correct, run the Check application definition (`CHKAPPDEFN`) command to highlight any problems.

### Physical security

Controlling physical access to your computer and workstations is the first defense against unauthorized access to your system.

### Password security

Require password changes frequently, and prohibit users from telling others their passwords. Use the password change procedures that are part of the operating system.

### Device security

Users can be authorized or excluded from specific workstations. Restrict QSECOFR to only the most essential workstations.

### Menu security

Use a menu management system to control which functions your users can perform when they sign on.

### Security level system value

Change the system security level (System Value QSECURITY) to "30" or higher. (Any value lower than "30" grants all users all rights to all objects.) Read the instructions in the IBM i Security Guide before doing so.

### Review special authorities

Set special authorities (those specified when you create a user profile) appropriately for each user. Review all users who have special authority other than \*USER and determine if the authority is appropriate for that user. You can display user profiles to a file and query that file.

## Review group profiles

Review all user profiles that have a group profile specified to determine if the group specified is correct. Review any user who is part of group QSECOFR, QPGMR, or QSYSOPR. These constitute a potentially serious security exposure. Review the authority for the group profiles to ensure that they are appropriate.

## Check adopted owner authority

Review all programs that adopt owner authority, especially those that adopt QSECOFR or other system user profiles.

Use the `DSPPGMADP` (Display Program Adopt) command to display programs that adopt owners, specifying user profile of owners of sensitive objects on your system. (Start with QSECOFR.)

## Check user profile menu

Check all user profiles that have a menu specified. Use `*SIGNOFF` for the menu parameter, except where necessary. If a user presses F3 from a main menu that is an initial program, the system will display the menu specified in the user profile.

## Output authority

Authority to report output should also be considered. Test authority to output queues and spool files to ensure that only authorized persons can view report or text output on your system.

---

## Questions

These are some of the questions you might be asked during a security audit.

### **Who has access to user profile QSECOFR, or to programs owned by QSECOFR that adopt owner's authority?**

If the number of people who have access to the QSECOFR profile is limited, then so is the likelihood that unauthorized access to programs and objects will occur. Only the owner of an object and people who use it should be granted authority to that object.

### **Who has \*ALLOBJ authority?**

Regular TURNOVER® Lifecycle Manager users should not have \*ALLOBJ authority. Even user profile TURNOVER does NOT need \*ALLOBJ authority for TURNOVER® Lifecycle Manager to work. In fact, no user other than QSECOFR needs \*ALLOBJ authority for TURNOVER® Lifecycle Manager to work, because strategic TURNOVER® Lifecycle Manager programs adopt QSECOFR authority.

### **Who has explicit authority to production data and program libraries?**

Regular TURNOVER® Lifecycle Manager users should not have explicit authority to production data and program libraries, and user profile TURNOVER does NOT need explicit authority for TURNOVER® Lifecycle Manager to work.

Many TURNOVER® Lifecycle Manager programs that are USRPRF(\*OWNER) are owned by TURNOVER® Lifecycle Manager, rather than QSECOFR. This is necessary so that TURNOVER® Lifecycle Manager can update the TURNOVER® Lifecycle Manager database as necessary without giving TURNOVER® Lifecycle Manager users explicit authority to the files (which would enable them to update the files by means other than TURNOVER® Lifecycle Manager).

If your auditors have a problem with user profile TURNOVER owning objects, then you can change user profile TURNOVER so as not to have a password (this will not cause any problems). However, the TURNOVER profile must have authority to update the TURNOVER® Lifecycle Manager files.

### **What programs on your system currently adopt owner's authority?**

You should audit any programs on your system that adopt owner's authority.

Do programs that adopt authority call other programs? If so, is the **Use Adopted Authority** parameter for the called program set to **No**?

USEADPAUT(\*YES) does *not* cause a program to adopt authority. It just means that if the program is called by another program which is USRPRF(\*OWNER), the adopted authority of that program is not dropped. In other words, if a program is USEADPAUT(\*YES) and it is owned by QSECOFR, and you call that program from a command line, you will *not* be adopting QSECOFR authority. USEADPAUT(\*YES) is the IBM-shipped default for the compile commands.

The TURNOVER® Lifecycle Manager programs that adopt QSECOFR authority all have their own internal authority checks to make sure the user running the program has the appropriate authorities defined within TURNOVER® Lifecycle Manager.

### **Do programs that adopt authority present a command line to the user? If so, what is the authority of the program owner?**

Anywhere that a user can enter a command in TURNOVER® Lifecycle Manager (that is, the Programmer Worklist command line, an F21 command line, or the command line in SEU) adopted authority is temporarily dropped (a program is in the stack that is USEADPAUT(\*NO)).

### **Who has access to the CHGPGM command, CHGOBJOWN command, and the CHGSYSLIBL command?**

| Command   | Description                                                                                                                              |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------|
| CHGPGM    | The User profile and Use Adopted Authority parameters of the CHGPGM command can be used to circumvent established security requirements. |
| CHGOBJOWN | This command can allow people to change object ownership, which can permit unauthorized access to objects.                               |



| Command    | Description                                                                                                                                                                                                                                        |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CHGSYSLIBL | If your programmers or users can modify their job's system library list, they could place a program higher in the library list than a product library. This creates a security exposure. IBM ships this command with authority limited to QSECOFR. |

## 10 Emergency changes

To perform emergency software changes, use one of these methods:

- create an emergency change application (this is the recommended method)

The emergency change application is usually a single-level application that allows you to make a change while someone else is working on an object. Those who are authorized to make emergency changes are usually restricted.

- use the multiple checkout feature.

The multiple checkout feature allows two programmers check out the same object at the same time, without defining a separate application.

---

### Using an emergency change application

You can define one or more emergency change applications. These enable you to check out source that is already checked out to someone else. You can change the source and test it in a separate development library prior to returning it to production.

When you check out source that has already been checked out to another programmer, **TURNOVER® Lifecycle Manager** submits a compare job to compare the original production source with the changed source in the development library. **TURNOVER® Lifecycle Manager** sends messages to the programmer who currently has source checked out, to the administrator of the emergency change application, and to `QSECOFR`.

When you view checked out source, you see a record of source checked out under both the emergency change application and the regular application. The **Emergency change** entry is highlighted to alert you that emergency changes are being processed.

#### To define an application as an emergency change application

On the Work with Application Definition panel, set **Emergency** to `Y`. For more information, see 'Working with application definitions' in the *TURNOVER® Lifecycle Manager User Guide*.

#### Recommendations

- Create one single-level emergency application definition to correspond with each of your applications. If all of your applications target the same production library, you can use just one emergency application definition to manage all of your emergency changes.

- Set up a special emergency change development library, with the appropriate source files, which are used exclusively to manage emergency changes. This eliminates the risk of overlaying any source that is already checked out.
- If you are using a source compare utility other than the one supplied by TURNOVER® Lifecycle Manager, set up the source compare defaults (option 11 on the Utility menu).

---

## Using the multiple checkout feature

The multiple checkout feature allows multiple users to check out the same object.

### To enable the multiple checkout feature

On the TURNOVER® Lifecycle Manager main menu, select option 8, then option 5. On the Set Global TURNOVER Defaults (SETGLOBE) panel, set **Allow group checkout** to `Y`.

When you do this, the **Exclusive lock** parameter appears on the Checkout command prompts. When you check out an object, you must specify whether or not you want to lock it:

- To prevent other users from checking out the same object, set **Exclusive lock** to `*YES`.
- To allow others to check out an object while you are working on it, set **Exclusive lock** to `*NO`.

On other panels, such as View Checked Out and Change Checkout, you can view and change this parameter.

If you try to check out something that someone else has checked out, you must confirm that you still want to check out the object. If you do, then all programmers who have it checked out receive a message informing them that you have checked it out.

When you promote an object, if any other programmers have promoted the same object *while* you had it checked out, you see a panel showing all of the objects on your form that have been changed while you were working on it. Before you can run the form, you must indicate that you have merged any changes the other programmers made into your version (see [‘Resolving concurrent changes’ on page 159](#)).

The system retains the fact that the programmer confirmed that they have reconciled the changes in this way. If you want a “Higher-authority” to confirm that all changes have been made, attach an approval list to your applications and ensure that the persons responsible for approving and/or running the form confirms that the changes have been made.

---

## Resolving concurrent changes

If concurrent changes result from either emergency change method (emergency change application or multiple checkout feature), you can resolve them.

### To view any objects that were changed concurrently by another programmer

- 1 Type `CHKCHGOBJ` on a command line, and then press Enter.

The Multiple Checkout Approval panel appears.

**Multiple Checkout Approval**
Your Company Inc.

The objects listed below have been turned over by other programmers while checked out to JIMCLARK. Changes made by those programmers must be included in changes made by JIMCLARK.

Position to object . . . . \_\_\_\_\_  
 Filter by programmer . . . . \_\_\_\_\_  
 Filter by reference . . . . \_\_\_\_\_

Type options, press Enter.

1=Approve    4=Cancel approval    5=View form    7=Project details

| Object    | Library | Type | Programmer | Form  | Date     | Run<br>Reference | Approved |
|-----------|---------|------|------------|-------|----------|------------------|----------|
| _ SLSF001 | SLSLIB  | RPG  | ABFORTIS   | 10431 | 04/30/20 | SLPR0321         | PSMITH   |

F3=Exit   F5=Refresh   F12=Cancel   F14=Display all

- 2 Selecting option 1 to confirm that the changes made by the other programmer have been made to your own version of the object.

### Notes

- This command is run automatically from the Work with Programmer Worklist panel whenever you build a form and, if not resolved then, again when you submit the form to run.
- A variant of this command is also run when you exit from editing a TURNOVER® Lifecycle Manager form and again during the pre-run error-check. When you have the multiple checkout feature enabled, TURNOVER® Lifecycle Manager does not run a form that contains objects you have not confirmed.
- If you try to promote an object to an intermediate level while another programmer's work is in process (at the same target level), TURNOVER® Lifecycle Manager prevents you from doing so. However, you can check out an object from an intermediate level. The multiple checkout rules apply to any level.
- If you are not sure whether your version of source matches another version of source, use the source compare command, CMPSRCMBR or select the **Compare source** option on the Programmer Worklist.

# 11 Managing multiple versions

The information in this section applies if you:

- support custom modifications for several customers or locations from a central location
- introduce changes to your software periodically, as releases
- maintain multiple national language versions of your software.

## About versions and releases

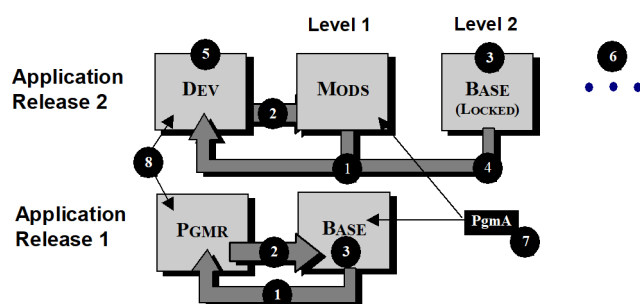
IBM i operating system is ideal for managing multiple releases of software, because of its library structure. As you start work on a new release, you create a new “release” library, start modifying objects from the last release and promote them to your new library. It does not matter whether you call the work in your new library a ‘Release’, a ‘Version’ or ‘Product/2100’. The important thing is that you intend to introduce all the changes to your application together.

An object version number distinguishes one version of an object from its predecessors or successors. It indicates how many times the object has been modified. Object version numbers help you verify what version of an object a user has installed.

## Related applications

TURNOVER® Lifecycle Manager manages multiple versions by using *related applications*.

In the diagram below, each box represents one or more libraries defined to that level of the application.



If you have to support more than one release, both of these applications might be active, simultaneously.

## How related applications work

- 1 Checkout is from right to left.
- 2 Promotions are from left to right.
- 3 The BASE application library is defined to each application.

The BASE library for Release 2 is locked. Changes are not promoted to locked levels for any application definition. (However, the BASE library can be changed using the Release 1 application.) MODS libraries contain only the objects that you have modified for that release.

Release 2 is defined as “related to” Release 1.

- 4 If, using Release 2, you check out an object that has not been changed before (and it therefore is not in the MOD library), TURNOVER® Lifecycle Manager looks for it in the locked libraries, defined at a higher level, starting from left to right.
- 5 If an object was already in development in a related application, you are informed of this “conflict” condition.
- 6 More than one locked level can be defined.

You must define application unlocked levels consistently across a set of related applications. That is, if you have one test level for the base application, you must have one test level for each related application.

- 7 Object names must be identical across related application boundaries.
- 8 When you check out an object from more than one related application, be sure the applications are set up so that they do not overlay each other in the same library.

In this example, the base application is set up to check out to the programmer’s library; the related application is set up to check out to a second library. Where many related applications are defined, define development libraries for each one.

## Additional rules

A *base* application definition defines the entire application, exclusive of those objects that have been modified for any location.

A *related* application consists of a *subset* of objects copied from the parent application and modified for use at one or more locations.

For each related application, the library list used to run the application will consist of the related application libraries followed by the Base application libraries. (In some instances, base objects may be merged into a new release library so that fewer libraries need to be in the library list.)

Objects and source that exist in the base may exist in one or more Related application libraries in their modified form. Object and source names must match in base and related libraries.

A Source member and object may exist in a Related application library without being present in the Base application library. These apply only to that version of the application. If such an object exists in several related application libraries, they must be changed independently. In order for TURNOVER® Lifecycle Manager to recognize objects as related, they must exist in the Base libraries.

Users who are authorized to the Base should also be authorized to each of the related applications. Users can be authorized to a related application without being authorized to the Base.

**Note** A related application could represent a release; a custom modification for a customer or location; a second language version of your application; or a combination of release, custom modification and language version.

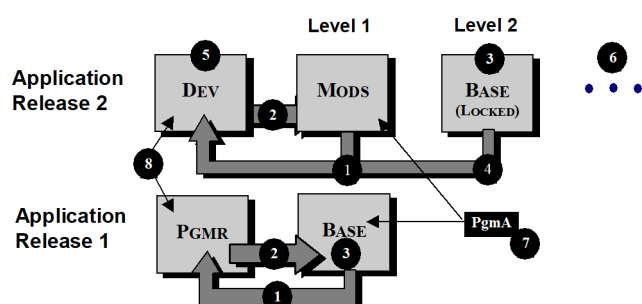
## Related applications and cross-referencing

The TURNOVER® Lifecycle Manager related application and cross-referencing features work closely together. If you check out a file from one application that is referenced by a program in another, whether it be defined as related or not, TURNOVER® Lifecycle Manager shows you the relationship. Although you can use the related application feature without building a cross-reference database, TURNOVER® Lifecycle Manager cannot predict the impact of your change. Therefore, you should do both. If you are managing multiple releases, create a new cross-reference table for each new release of your application.

## Recommendation

If the related application represents customer modifications and you distribute changes to customers, keep the customer modifications in separate libraries from the base application on the customer's computer. If changes are installed into one, consolidated library on your customer's computer, they must be installed in the correct order to avoid overlaying customized objects with those from the base application. TURNOVER® Lifecycle Manager handles the installation correctly. Normally, all changes are run in the same Group Form job, so this is not a problem. However, if you distribute and apply changes manually, they could be installed in the wrong order.

## Related application change scenarios



## A change is made to an (older) program in Release 1

If the change is in MODS (in Release 2), you might have to make the same change there too.

You can check it out from Release 1 and Release 2. TURNOVER® Lifecycle Manager prompts you to do so.

If you do *not* check the program out of MODS, TURNOVER® Lifecycle Manager considers that a conflict and does not allow you to promote the object to BASE without verifying that you have resolved the conflict (see *'Resolving version conflicts' on page 168*).

If the program is already checked out of MODS, then a conflict exists that you must resolve: you must either make a similar change to the related object or explain why you did not (for example, it might have been completely rewritten for Release 2).

If the program is not in MODS, you check it out, change it and promote it using just the Release 1 application.

## A change is made to a physical file for Release 2

(Assuming you use TURNOVER® Lifecycle Manager's own cross-reference.)

If the file is not in the MOD library, it is checked out and copied from the Base library, and then promoted back to the MOD library.

If the file is in the MOD library, it is checked out of the MOD library.

If logical files exist over the physical file, but are not changed, they are created in the MOD library.

A program that uses the physical file and does not exist in the MOD library will be created there from the source in the BASE.

**Note** The cross-reference file is built in a table for Release 2 using all the application libraries, including those in the locked levels.

## A change is made to a file for Release 1

If the file has been modified and exists in Release 2, you are prompted to check it out. If you do not, you must state why, to avoid a conflict.

If programs exist over the physical file in either Release 1 or 2, they are recompiled when you promote the file to the base.

Each application uses its own cross-reference table and recompiles the correct related objects for each release.

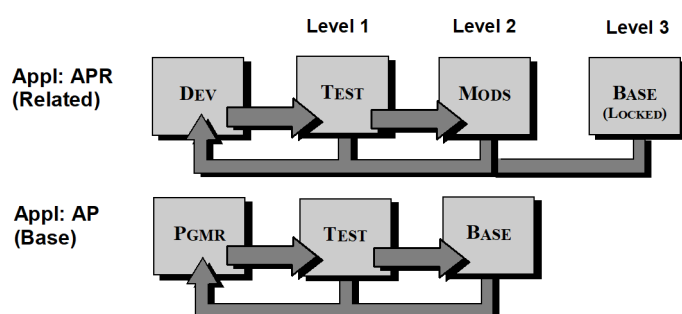


## Setting up application relationships

You can interrelate application definitions so that when you change one application, TURNOVER® Lifecycle Manager prompts you to check if a similar change is required for a related application. In application relationships, the base application is referred to as the *parent* application; a related application is referred to as a *child* application.

Before you can set up application relationships, you must first define each of the applications independently (see 'Working with Application Definitions' in the TURNOVER® Lifecycle Manager User Guide). Each application must be defined similarly.

### Example



This is similar to the example in 'Related application change scenarios' on page 163, except that there are two unlocked levels, levels 1 and 2.

### To set up application relationships

On the Work with Application Definitions panel, select an application definition with option 19 (Application relations).

The Work with Application Relationships appears.

| MM/DD/YY                                                                           | Work with Application Relationships |     |                              |                                   | Your Company, Inc. |
|------------------------------------------------------------------------------------|-------------------------------------|-----|------------------------------|-----------------------------------|--------------------|
| Selected application . . . <b>ARA Accounts Receivable Application</b>              |                                     |     |                              |                                   |                    |
| Release . . . . .                                                                  |                                     |     |                              |                                   |                    |
| Version . . . . .                                                                  |                                     |     |                              |                                   |                    |
| Parent application . . . <b>APA Accounts Payable Application</b>                   |                                     |     |                              |                                   |                    |
| Release . . . . .                                                                  |                                     |     |                              |                                   |                    |
| Version . . . . .                                                                  |                                     |     |                              |                                   |                    |
| The following applications have child relationships with the selected application: |                                     |     |                              |                                   |                    |
| 1=Add child 2=Change rule 4=Remove child relationship                              |                                     |     |                              |                                   |                    |
| New Obj.                                                                           |                                     |     |                              |                                   |                    |
| Appl                                                                               | Rel                                 | Ver | Rule                         | Description                       |                    |
| —                                                                                  | —                                   | —   | *NONE                        |                                   |                    |
| —                                                                                  | ARAB                                |     |                              | Accounts Receivable B Application |                    |
| —                                                                                  | ARAC                                |     |                              | Accounts Receivable C Application |                    |
|                                                                                    |                                     |     |                              |                                   | Bottom             |
| F3=Exit F4=Select child                                                            |                                     |     | F12=Cancel F23=Remove parent |                                   |                    |

This panel shows all application relationships for the application you selected. Its child relationships are listed at the bottom of the panel (in this example, ARAB and ARAC).

## Nesting related applications

The base application definition is the *parent*. It defines the entire application, exclusive of those objects that have been modified for any location. A related application is the *child*. It consists of a *subset* of objects copied from the parent application and modified for use at one or more locations.

An application can be a parent of one application, and the child of another. This is called a *nested* related applications. For example (see '[Related application change scenarios](#)' on page 163), you might have several customer modifications applications related to Release 1 and several others related to Release 2. In that case, the Release 2 application would have several other applications related to it.

For the example shown above, it is likely that you would start a Release 3 someday. That would be related to Release 2.

For an illustration of a nested application, see '[Advanced related applications](#)' on page 172.

## Using related applications

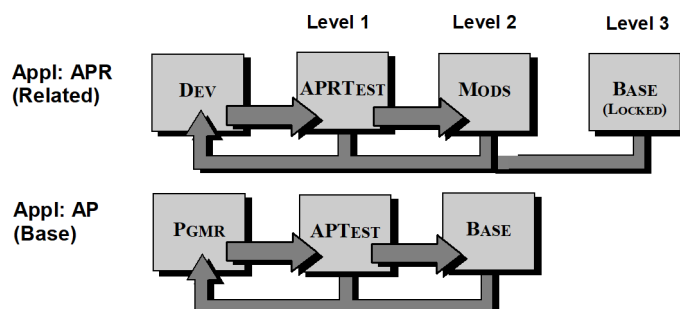
Related applications apply:

- when you check out source for an object
- when you set up TURNOVER® Lifecycle Manager forms to promote your changes.

## Working with multiple versions

The Programmer's Worklist enables you to work on all of the related objects across all of your related applications at one time.

In the example below, when you check out an item from the AP BASE library, TURNOVER® Lifecycle Manager searches for the same object in the MODS library and APRTTEST. It also tells you if it is checked out for the APR application.



You can check it out for AP and APR. If it is already checked out for APR and has not been promoted to APRTTEST, TURNOVER® Lifecycle Manager writes a conflict record.

If you ignore a related item and do not check it out, another conflict record is written. You must resolve the conflict before you can promote the object to the Base application. (See [‘Resolving version conflicts’ on page 168.](#))

All of this is managed from the Programmer’s Worklist panel.

### Example Programmer’s Worklist panels

```
MM/DD/YY 12:25:35      Work with Programmer Worklist      Your Company, Inc.

Worklist . . . . APPR0001  A test task
Position to . . . .      Apply filters . . . N (F17=Filters)

2=Change 3=Copy 4=Delete 11=Move 12=Compare source 99=Next option
21=Checkout 22=Change checkout 24=Checkin 25=Task details 26=Conflicts

      Object      Attr  Library  Obj Src Out  Form  FormSts  ItemSts  Nxt
      _____  _____  _____  _____  _____  _____  _____  _____
      Object      Attr  Library  Obj Src Out  Form  FormSts  ItemSts  Nxt
      _____  _____  _____  _____  _____  _____  _____  _____
      _____  _____  _____  _____  _____  _____  _____  _____
      _____  _____  _____  _____  _____  _____  _____  _____

Parameters or command
====>

F2=Forms F3=Exit F4=Prompt F5=Refresh F6=Add item F9=Retrieve
F10=Command entry F11=View2 F19=Inter. 21/46 F23=More options F24=More keys
```

| To                                                 | Do this        |
|----------------------------------------------------|----------------|
| search for related objects in related applications | Use option 16. |
| show all conflicts for the object selected         | Use option 26. |

Cross-referencing (option 15) works in a similar way.

## Searching for related objects in related applications

When you select option 16, the Work with Related Applications panel appears.

```
MM/DD/YY 12:27:55      Work with Related Applications      Your Company, Inc.

Worklist . . . . TP400257  Tasks-No validity chking on filter value

1=Add to Worklist 2=Add to Worklist/Checkout 8=Browse base 9=Browse related

      Member      Attr  -----Base-----  ----Related-----  Chckd
      Member      Attr  Appl Rel Ver Lev  Appl Rel Ver Lev  Out Programmer
      _____  _____  _____  _____  _____  _____  _____  _____  _____  _____
      _____  _____  _____  _____  _____  _____  _____  _____  _____  _____
      _____  _____  _____  _____  _____  _____  _____  _____  _____  _____

Bottom
F3=Exit F4=Prompt F11=Alt. view F12=Cancel F19=Submit 1/2 F21=Commands
```

This panel shows all versions of the objects you have modified in related applications (subsequent releases, foreign language versions, customer modifications, and so on.)

In the panel above, two versions exist, one for related application APR and one for APR2. The second one is currently checked out.

If you check out the item in the base application on the worklist, but do not check out the related objects, TURNOVER® Lifecycle Manager writes a conflict record for each of the related applications where the object has been changed.

If the object in the related application is already checked out, the programmer who checked it out must resolve the conflict before creating a TURNOVER® Lifecycle Manager form.

---

## Resolving version conflicts

If a conflict exists, the Work with Programmer Worklist panel displays **CONFLICT** in the **ItemSts** field for the item.

|                                                                            |      |                                     |     |     |     |                    |         |          |         |
|----------------------------------------------------------------------------|------|-------------------------------------|-----|-----|-----|--------------------|---------|----------|---------|
| MM/DD/YY 10:05:32                                                          |      | Work with Programmer Worklist       |     |     |     | Your Company, Inc. |         |          |         |
| Worklist . . . . TEST                                                      |      | Test worklist                       |     |     |     |                    |         |          |         |
| Position to . . . .                                                        |      | Apply filters . . . Y (F17=Filters) |     |     |     |                    |         |          |         |
| 2=Change 3=Copy 4=Delete 11=Move 12=Compare source 99=Next option          |      |                                     |     |     |     |                    |         |          |         |
| 21=Checkout 22=Change checkout 24=Checkin 25=Task details 26=Conflicts     |      |                                     |     |     |     |                    |         |          |         |
| Object Object Chk                                                          |      |                                     |     |     |     |                    |         |          |         |
| Object                                                                     | Attr | Library                             | Obj | Src | Out | Form               | FormSts | ItemSts  | Opt     |
| APCLP001                                                                   | CLP  | PRODTEST                            | Y   | Y   | Y   |                    |         | NEW      |         |
| APCLP001                                                                   | CLP  | ACPTTEST                            | Y   | Y   | N   |                    |         | NEW      | 47      |
| APCLP001                                                                   | CLP  | DEVTEST                             | N   | Y   | N   |                    |         | NEW      | 32      |
| 26 APCLP002                                                                | CLP  | PRODTEST                            | Y   | Y   | Y   |                    |         | CONFLICT |         |
| APCLP002                                                                   | CLP  | ACPTTEST                            | Y   | Y   | N   |                    |         | NEW      | 47      |
| APCLP002                                                                   | CLP  | DEVTEST                             | N   | Y   | N   |                    |         | NEW      | 32      |
| APRPG001                                                                   | RPG  | PRODTEST                            | Y   | Y   | Y   |                    |         | NEW      |         |
| APRPG001                                                                   | RPG  | ACPTTEST                            | N   | N   | N   |                    |         | NEW      | 47      |
| APRPG001                                                                   | RPG  | DEVTEST                             | N   | Y   | N   |                    |         | NEW      | 32      |
|                                                                            |      |                                     |     |     |     |                    |         |          | More... |
| Parameters or command                                                      |      |                                     |     |     |     |                    |         |          |         |
| ====>                                                                      |      |                                     |     |     |     |                    |         |          |         |
| F2=Forms F3=Exit F4=Prompt F5=Refresh F6=Add item F9=Retrieve              |      |                                     |     |     |     |                    |         |          |         |
| F10=Command entry F11=View2 F19=Batch 21/46 F23=More options F24=More keys |      |                                     |     |     |     |                    |         |          |         |

To resolve the conflict, you can either check out the related item or, if you decide that the related item does not need to be changed, you can update the conflict file. To do so, type 26 next to the item, and then press Enter.

The Work with Related Application Conflicts panel appears.

Work with Related Application Conflicts panel

MM/DD/YY  
10:22:15  
Position to object . . . \_\_\_\_\_

Work with Related Application Conflicts

Your Company, Inc.

Highlighted objects indicate unresolved conflicts.

1=Send message    2=Change    5=Display    8=Browse base    9=Browse related

|          |      |      | -----Base----- |     |     |      | -----Related----- |     |     |      |          |  |
|----------|------|------|----------------|-----|-----|------|-------------------|-----|-----|------|----------|--|
| Object   | Type | Appl | Rel            | Ver | Lev | Appl | Rel               | Ver | Lev | Code | Comment  |  |
| APCLP002 | CLP  | AP   | 0              | 0   | 2   | APR  | 0                 | 0   | 2   | NA   | Obsolete |  |
| APCLP002 | CLP  | AP   | 0              | 0   | 2   | APR2 | 0                 | 0   | 2   |      |          |  |

F3=Exit    F11=Alt. view    F12=Cancel    F17=Filters

Bottom

This panel shows you all of the resolved or unresolved entries in the conflicts file for this item.

| To                                                      | Do this                                                                 |
|---------------------------------------------------------|-------------------------------------------------------------------------|
| send a message to the user associated with the conflict | Use option 1.                                                           |
| browse the base source member                           | Use option 8.                                                           |
| browse the related source member                        | Use option 9.                                                           |
| resolve a conflict for an item                          | Type 2 next to the item. The Change Conflict panel appears (see below). |
| display other views                                     | Press F11.                                                              |

## Change Conflict panel

|                              |                |                        |                |                    |  |
|------------------------------|----------------|------------------------|----------------|--------------------|--|
| MM/DD/YY 10:25:39            |                | <b>Change Conflict</b> |                | Your Company, Inc. |  |
| Type choices, press Enter.   |                |                        |                |                    |  |
| Object . . . . .             | PROGRAMX       |                        |                |                    |  |
| Object type . . . . .        | RPG            |                        |                |                    |  |
| Conflict type . . . . .      | *CHKOUT        |                        |                |                    |  |
|                              | <u>Base</u>    |                        | <u>Related</u> |                    |  |
| Application . . . . .        | AP             |                        | APF            |                    |  |
| Release . . . . .            | 3              |                        | 3              |                    |  |
| Version . . . . .            | 0              |                        | 0              |                    |  |
| Level . . . . .              | 2              |                        | 2              |                    |  |
| Library . . . . .            | APPROD         |                        | APPRODR        |                    |  |
| Time stamp . . . . .         | 19960219142345 |                        |                |                    |  |
| User . . . . .               | PGMR1234       |                        | PSCHMIDT       |                    |  |
| Project/task reference . . . | APPR0023       |                        | APPR0023       |                    |  |
| Form . . . . .               | 1000342        |                        |                |                    |  |
| Reason code . . . . .        | _ F4=List      |                        |                |                    |  |
| Comment . . . . .            |                |                        |                |                    |  |
| F3=Exit F4=List              |                | F9=Base task           |                | F12=Cancel         |  |

You can resolve a conflict by entering a reason code.

| To                                                        | Do this                           |
|-----------------------------------------------------------|-----------------------------------|
| list the reason codes                                     | Press F4 to list the valid codes. |
| see the project and task associated with the base changes | Press F9.                         |

**Note** Resolve a conflict only if you have resolved it. If the related program needs to be changed, add it to your worklist, check it out, and then change it. When you check it out, TURNOVER® Lifecycle Manager updates the conflict file.

### Audit history of conflict changes

To view and resolve the conflicts, use the **TWRKCON** (Work with Conflicts) command

To produce a report of related application conflicts, use the **TRPTCON** (Report Conflicts) command.

## Conditions for which conflicts may exist

| Condition                                                                                 | Action                                        |
|-------------------------------------------------------------------------------------------|-----------------------------------------------|
| You checked out an object from the base application but not from the related applications | Add object to your Worklist and check it out. |

| Condition                                                                                               | Action                                                                                                                                        |
|---------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| The item in the related application is checked out to someone else                                      | Use worklist option 26 to send a message to the other programmer. They must resolve the conflict before the item in the base can be promoted. |
| You do not have time to checkout and fix the related item at this time                                  | Resolve with an appropriate Reason code, and then permanently resolve the conflict later.                                                     |
| The related item was checked out but then you realized that it did not need to be changed.              | Check in the item.<br><br>The conflict file is updated to indicate that the item was checked in manually.                                     |
| The object in the base application has been promoted, but the ones in the related applications has not. | Do nothing.<br><br>If the item is checked out, it will be resolved automatically when it is promoted.                                         |

## Building a form with unresolved conflicts

When objects in the related application were created from the source in the base, there is no conflict. (For example, if you might maintain different national language versions of objects from the same source and different language message files located in related application libraries).

- If the Programmer Worklist session default, **Related apps check automatic**, is set to Y, these objects are included for recompile on TURNOVER® Lifecycle Manager forms (See Programmer's Worklist, F18=Defaults.)
- If **Related apps check automatic** is set to N, the Form Maintenance Options panel appears when you build the form from your Worklist (option 46) or if you edit the form (option 42).

|                                                                      |                                 |                    |
|----------------------------------------------------------------------|---------------------------------|--------------------|
| MM/DD/YY<br>11:10:12                                                 | <b>Form Maintenance Options</b> | Your Company, Inc. |
| Type choices, press Enter.                                           |                                 |                    |
| Form . . . . . 1000076                                               |                                 |                    |
| Description . . . . . Test worklist                                  |                                 |                    |
| Resequence the form . . . . . <u>Y</u> Y, N                          |                                 |                    |
| Check for logical files and add them to the form . . . <u>Y</u> Y, N |                                 |                    |
| Check cross-reference file for related objects . . . . <u>Y</u> Y, N |                                 |                    |
| Check for objects changed by other programmers . . . . <u>N</u> Y, N |                                 |                    |
| Check for related application conflicts . . . . . <u>Y</u> Y, N      |                                 |                    |
| F3=Exit F12=Cancel                                                   |                                 |                    |

When you press Enter, and if conflicts exist for items in applications related to those on the TURNOVER® Lifecycle Manager form, the Work with Conflict panel appears (see *'Work with Related Application Conflicts panel' on page 169*).

## Approving forms with version conflicts

When you approve a form (option 48 on the Programmer's Worklist or option 10 on the Work with Forms panel) for which related applications exist, you can display any conflicts by pressing F8.

## Submitting a form with unresolved conflicts

If you try to submit a form with unresolved conflicts, an error message appears.

```
MM/DD/YY                      Submit Form                    Your Company, Inc.  
11:22:24                                         SYSTEM: SLS  
  
Form . . . . . 1000076      Test worklist  
Application . . . . . AP      Release:       Version:       Level:   1  
Status . . . . . READY  
  
Type options, press Enter.  
    Form schedule date . . . . . *CURRENT *CURRENT, date  
    Form schedule time . . . . . *CURRENT *CURRENT, hhmmss  
    Submit on hold . . . . . *NO          *YES, *NO  
  
. . . . .  
:  
:  
: Error checking completed with the following warning and error messages: :  
: Error: Related application conflicts found for CLP APCLP002. :  
:  
:  
:  
:  
:  
:  
:  
:  
:F12=Cancel form submit :  
:  
:. . . . .  
Error checking has completed for form.
```

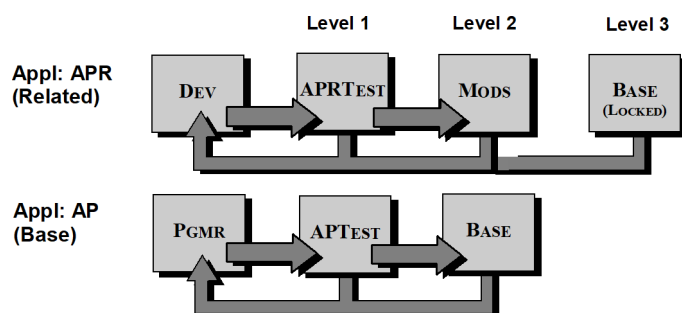
To resolve the conflict, cancel the submit request and resolve the conflict. Conflicts are recorded only if an object of the same name as the one you changed was found in a related application, not for every object in the base application. If there are Error messages, the form does not run.

## Advanced related applications

## Managing customized versions of software

If you maintain different customized changes for each of your customers, customized changes for several different plant locations, using the same software, you might define and relate the applications like this:





When you check out an item for the Base application, you need to know if any of the items you are checking out were customized for either location; in other words, have the objects been modified for either location 1 or 2?

This example shows only two related applications in this example, but there is really no limit to the number of related applications you can define: you might have several hundred sets of custom modifications.

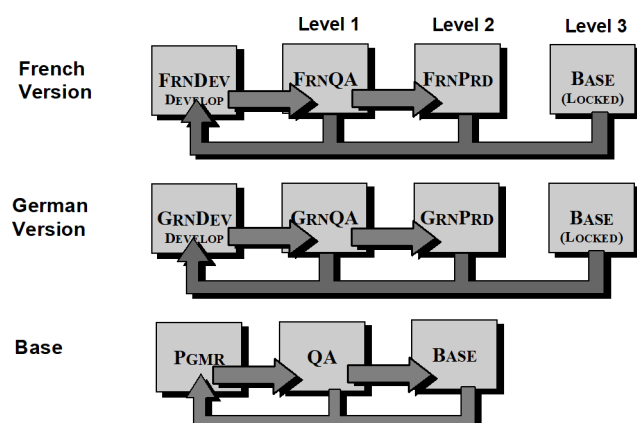
## Managing multiple language versions of software

The most efficient method of creating different language versions of a software package is to put the national language text into message files and then translate the message file text.

On the IBM i computer, display files can be defined to allow the text to be loaded at run time. Status and error messages are in the same way.

Commands and printer files have embedded text strings, but they can also have externalized language text. However, they must be compiled over the correct national language message file.

No matter how many language versions of software you have, you only need *one* copy of source. To maintain different language versions of your software which are otherwise identical, you might to set up the applications like this:



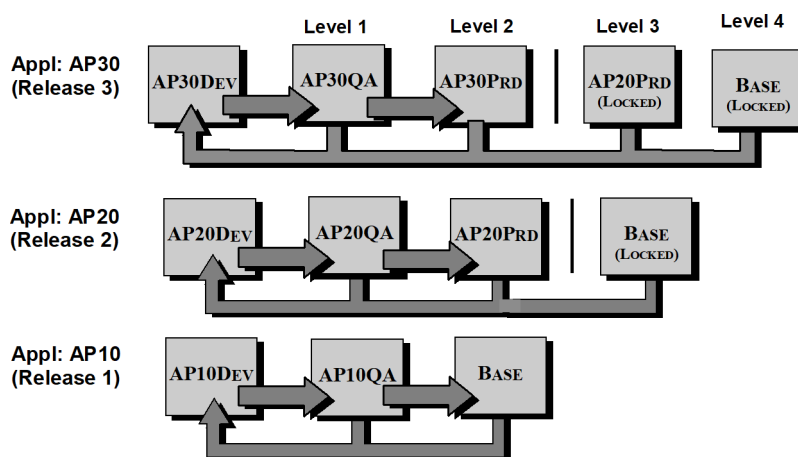
In these applications, the GRNPRD and FRNPRD libraries contain a message file and just those objects that need to be compiled over it, such as commands and printer files

Other than maintaining the translated versions of the message file, no changes need to be made directly to these objects. Changes only need to be made to the source in the base application; in this example, *no source exists in the language libraries*. When you build a Base application form to promote any language-sensitive objects from, say, QA to BASE, TURNOVER® Lifecycle Manager creates forms for each of the related language applications. When these run, TURNOVER® Lifecycle Manager creates the objects in the appropriate language library for each related application using the source from the base application compiled over the appropriate language message file. Your application must manage the substitution of the correct message file and language library when it is running. (See '[Application planning](#)' on page 10.)

For information about managing multiple language versions of several releases, see '[Example application definition schematics \(examples\)](#)' on page 175.

## Managing multiple releases of software

This example manages three releases, which need to supported simultaneously.



In this example, Release 1 is the Base for Release 2 and Release 2 is the base for Release 3. This is called a nested related application.

If you find an error, it is likely that you would want to check out and fix all copies of the program affected.

- 1 Start by identifying the objects you need from the *oldest* release.
- 2 Add these to your Programmer's worklist, and then select option 16 to check for related objects. If you need to fix these, add them to your Worklist. Then check them all out, using option 21.
- 3 Once you have checked them out (into different libraries), change and test each version. Then promote them all back.

What if another programmer is modifying a program for Release 3 and you need to check out the same object for Release 1, 2, and 3?

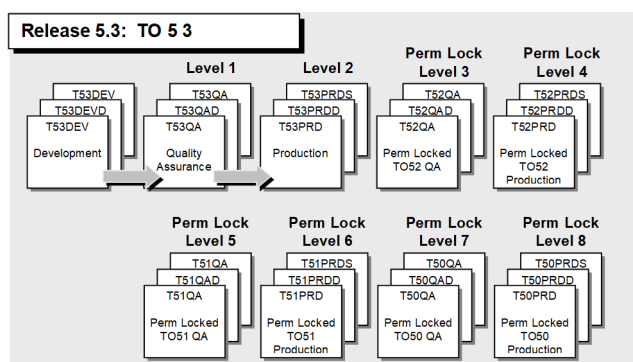
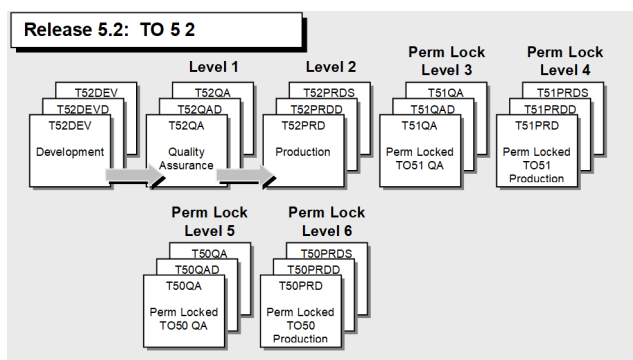
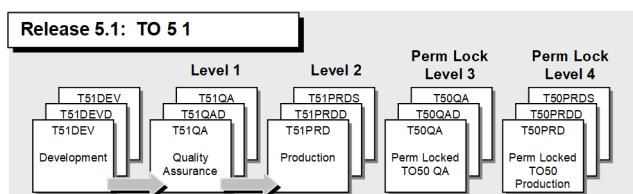
- You can wait until they promote the object back to production (AP30Prd in the diagram).

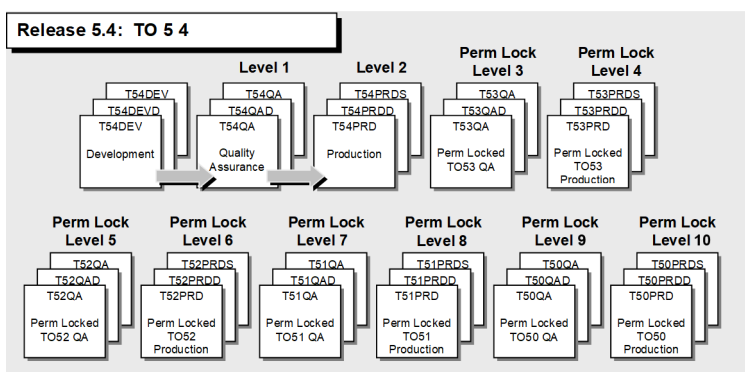
- You can check out the object for Release 1 and 2 and put the third one on your Worklist, with the intention of changing it later.
- You can message the programmer, and ask them to fix the Release 3 version of the object.
- You can *not* just forget it and not fix the Release 3 version of the object. You must resolve the conflict before promoting the Release 1 and 2 versions.

If the other programmer has already promoted the object to the test level (AP30QA), it is still checked out of AP30PRD, so a conflict record is written.

## Combining multiple releases with language or versions (or both)

### Example application definition schematics (examples)





The oldest supported version, Release 5.1, is the base for Release 5.2; Release 5.2 is the base for 5.3; Release 5.3 is the base for 5.4 and so on.

All of the *objects* that are in the base release are also in subsequent releases (unless they were intentionally removed from the product). However, source code exists in subsequent releases *only if the object was changed in that release*. If you check out a source member for Release 5.1, TURNOVER® Lifecycle Manager shows you the related source in subsequent releases only if it exists there (that is, if it was changed for both releases). If an object is changed for Release 5.1 but not in subsequent releases, then the object is automatically recompiled into subsequent Releases (5.2 and 5.3) from the source in Release 5.1.

If a new object is created in Release 5.1, TURNOVER® Lifecycle Manager creates the object in subsequent releases, depending on the new related object rule setting in the related application file (TAPPRELF). You can set the object rule for the child relationships in an application on the Work with Application Relationships panel. Also, you can use the New Related Object (TNEWRELOBJ) command to change the rule for any related application.

The possible settings for the new related object rule are as follows:

- 1 = Do not create new objects in subsequent releases.
- 2 = Create data objects in subsequent releases.
- 3 = Create all objects in subsequent releases.

Most companies managing multiple releases use option 3.

## Object versioning

Application versioning and object versioning are different, but similar.

*Application versioning* means using related applications along with the native IBM i library management capabilities to manage a set of related objects.

*Object versioning* helps you to identify:

- the version of a specific object, by distinguishing it from its predecessors and successors
- what version of an object a user is using

- how an object was changed, who changed it, why it was changed, and so on, *every time it was changed*, throughout the object's life.

IBM i stores some of this information its stores in its object description, and TURNOVER® Lifecycle Manager stores more information.

TURNOVER® Lifecycle Manager maintains a version number for all objects you maintain using it.

## Version and Object Stamping Defaults panel

| MM/DD/YY                                                                  | Version and Object Stamping Defaults |      |      |        | Your Computer, Inc. |
|---------------------------------------------------------------------------|--------------------------------------|------|------|--------|---------------------|
| 16:09:22                                                                  | Application: AP                      | Rel: | Ver: | Lev: 2 | SYSTEM: SLS         |
| Type choices, press Enter.                                                |                                      |      |      |        |                     |
| <b>Version Control Settings</b>                                           |                                      |      |      |        |                     |
| Version number to increment on promotion . . *MAJOR *MAJOR, *MINOR, *NONE |                                      |      |      |        |                     |
| <b>Object Stamping Settings</b>                                           |                                      |      |      |        |                     |
| Stamp object description . . . . . Y Y, N                                 |                                      |      |      |        |                     |
| <b>Values to stamp:</b>                                                   |                                      |      |      |        |                     |
| Object version number . . . . . Y Y, N                                    |                                      |      |      |        |                     |
| Form number . . . . . Y Y, N                                              |                                      |      |      |        |                     |
| Project/task reference . . . . . Y Y, N                                   |                                      |      |      |        |                     |
| TurnOver type code . . . . . Y Y, N                                       |                                      |      |      |        |                     |
| F3=Exit F12=Cancel                                                        |                                      |      |      |        |                     |

### Version Control Settings

Every object that is changed by TURNOVER® Lifecycle Manager has a major and minor version number associated with it. The format is *xxx.yyy*, where *xxx* is the major version number, and *yyy* is the minor version number, for example 17.2.

When an object is promoted for a given level, you can specify whether to increment the major or minor version number.

Typically, you increment the major version number when the object is promoted to the highest unlocked level (production), and you increment the minor version number when the object is promoted to lower levels.

When the major version is incremented, the minor version is reset to zero.

### Object Stamping Settings

If you set the **Stamp object description** parameter to Y, and one or more parameters (**Object version number**, **Form number**, **Project/task reference**, and **TURNOVER® Lifecycle Manager type code**), to Y, TURNOVER® Lifecycle Manager updates the object description with the corresponding information about the object.

If you plan to distribute objects for an application definition that uses MO (Move Object) or CD (Create Duplicate) as its promotion method, make sure that object stamping is turned off in the remote application definition.

**Note** TURNOVER® Lifecycle Manager information is mapped into object description fields wherever space is available: fields not necessarily descriptive of the data stored in them. The `TDSPOBJD` command (see [‘TDSPOBJD command: Display Object Description’ on page 179](#)) displays this information.

### Object version number

`Y` stores the `xxx.yyy` version number of the object in the licensed program portion of the object description.

To distinguish between changes made to the same object in different releases of the same application, the Release and Version of the application used to promote the object are stored in this field. The application code associated with the object is stored in the first four characters of the APAR ID field in the object description. This information, and the release and version, is useful when you have multiple applications with the same release and version number, such as when you support multiple languages.

### Form number

`Y` stores the form number of the form that created the object in the PTF number field of the object description.

### Project/task reference

`Y` stores the first eight characters for the project/task reference of the last change to this object in the Object control level field of the object description. The last two characters are stored in the last two characters of the APAR field of the object description.

### TURNOVER® Lifecycle Manager type code

`Y` stores the TURNOVER® Lifecycle Manager type code used the last time the object was changed in the user-defined attribute field of the object description. TURNOVER® Lifecycle Manager checkout programs retrieve this type code when checking out an object by retrieving it from this field in the object description, if available.

---

## Excluding objects from stamping

You might need to exclude certain objects from being stamped in the object description. The `TCODOMTF` (Change Object Description Omit) file has a 1 character field for each of the values (object version number, form number, project/task reference, and type code) that you can stamp in the object description. You can exclude any or all of these values from stamping by adding them the `TCODOMTF` file and placing an `X` in the appropriate field. You can specify the library or use `*ALL` to exclude object stamping for all libraries.

---

## Sample display of object description

When you run the `DSPOBJD` command (`*SERVICE`) for an object after it was promoted, the Display Object Description panel appears. The [‘TDSPOBJD command: Display Object Description’ on page 179](#) formats this information.

| Display Object Description - Service |   |                   | Library 1 of 1 |
|--------------------------------------|---|-------------------|----------------|
| Object . . . . .                     | : | TESTCLP           |                |
| Library . . . . .                    | : | MARKPHIP          |                |
| Library ASP device . . . . .         | : | *SYSBAS           |                |
| Type . . . . .                       | : | *PGM              |                |
| Source file . . . . .                | : | QCLSRC            |                |
| Library . . . . .                    | : | MARKPHIP          |                |
| Member . . . . .                     | : | TESTCLP           |                |
| Attribute . . . . .                  | : | CLPI              |                |
| User-defined attribute . . . . .     | : |                   |                |
| Freed . . . . .                      | : | NO                |                |
| Size . . . . .                       | : | 36864             |                |
| Creation date/time . . . . .         | : | 03/23/04 11:32:30 |                |
| Source file date/time . . . . .      | : | 03/23/04 09:27:40 |                |
| System level . . . . .               | : | V5R2M0            |                |
| Compiler . . . . .                   | : |                   |                |
| Object control level . . . . .       | : | TP400231          |                |
|                                      |   |                   | More...        |
| Press Enter to continue.             |   |                   |                |
| F3=Exit F12=Cancel                   |   |                   |                |

Page down to see more information.

| Display Object Description - Service |   |               | Library 1 of 1 |
|--------------------------------------|---|---------------|----------------|
| Object . . . . .                     | : | TESTCLP       |                |
| Library . . . . .                    | : | MARKPHIP      |                |
| Library ASP device . . . . .         | : | *SYSBAS       |                |
| Type . . . . .                       | : | *PGM          |                |
| Changed by program . . . . .         | : | NO            |                |
| User modified . . . . .              | : | YES           |                |
| Licensed program . . . . .           | : | 007.008 14.11 |                |
| PTF number . . . . .                 | : | 0104232       |                |
| APAR ID . . . . .                    | : | AP            |                |
| Text . . . . .                       | : | Test program  |                |
|                                      |   |               | Bottom         |
| Press Enter to continue.             |   |               |                |
| F3=Exit F12=Cancel                   |   |               |                |

## TDSPOBJD command: Display Object Description

The TDSPOBJD command can be duplicated into a library in the system part of your library list and used as an enhanced version of the DSPOBJD command. This command recognizes those fields in the object description that TURNOVER® Lifecycle Manager updates.

```
MM/DD/YY 16:32:46 Display Object Description - TURNOVER® Your Company, Inc.

Object Information:
Object . . . . . : RPG001          Object version . . . . : 003.000
Library . . . . . : MPPROD         Appl/Rel/Ver . . . . : AP/00/00
Type . . . . . : *PGM
Object owner . . . . : FISHER
Creation Date/Time . . : 12/29/95 15:40:20
Text . . . . . : Reads Record in CONTPF

Source Information:
Source file . . . . . : QRPGLSRC
Library . . . . . : MPPROD
Member . . . . . : RPG001
Attribute . . . . . : RPGLE
Source file date/time . : 12/29/95 15:39:36

TurnOver Information:
Created by form . . . . : 1000028    Change RPG001 to Order Header File
Project/task reference . : MPTP0001   Change RPG001 to Order Header File
Type code . . . . . : RPGLE

Bottom
F3=Exit F5=*FULL F6=View form F8=*SERVICE F9=View task F12=Cancel
```

This panel displays the same information the IBM i display object description command, and the TURNOVER® Lifecycle Manager audit information.

To view the form or project task record for an object, press F6 or F9.

This command also displays the object's version number.

To access any of the other information in the object description, there are function keys that run the IBM i DSPOBJD command.

## Viewing object version on Work with Object History

Option 11 (Line Details) on the Work with Object History panel displays the version numbers of the object you selected. This enables you to track the version history, and display the version number for non-IBM i object types (which cannot be stamped as the native IBM i objects can), such as copybook code, MSG ID's, \*DATA files, and so on.

---

## TSETAPPOBJ command: Set Application Object

The TSETAPPOBJ command sets the initial version number for all existing objects in an application to a value you specify.

TURNOVER® Lifecycle Manager stores an object's version information in the licensed program portion of the object description. To see how the object version information appears in the object description, see *'Sample display of object description' on page 178*.

If you have been using earlier versions of TURNOVER® prior to these new capabilities, TSETAPPOBJ gets the object history database to determine the last form number, project reference and type code for an object, and update those values in the object description.

TSETAPPOBJ also enables you to process just those objects for an application that have not already been processed. This is useful if you add a new library to an existing application, and you just want to stamp the new objects.



## TSETOBJVER command: Set Object Version

This command is the same as the TSETAPPOBJ command, except it processes objects by name, library and type. This makes it useful for processing subsets of objects in an application.

The TSETAPPOBJ command uses the TSETOBJVER command after first determining the libraries and objects in an application from the TURNOVER® Lifecycle Manager database.

### Setting object values during the initial build

The TURNOVER® Lifecycle Manager initial build process sets the version number to 1.0 for all objects not already defined to TURNOVER® Lifecycle Manager, and updates the object description based on the application settings. If you currently use one of the user fields in the object description, define your application (see *'Setting up application relationships' on page 165*) such that you avoid overlaying the fields you use.

## Logic used to determine version number

### 1 Is Version field in line record filled in?

- |     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Yes | Use this as base for incrementing version number based on application. This will only be the case in two scenarios: <ul style="list-style-type: none"> <li>▪ Distribution: This field will be filled in with the version number that was set on the development system. Presumably, you will choose to increment with *NONE on the remote system to keep the version numbers the same on all systems.</li> <li>▪ Set by exit program: A user may choose to preset this value with an exit program which runs before the version is set, such as the pre-run or add line exit.</li> </ul> |
| No  | Continue.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

### 2 If Version Increment Value is \*MINOR or \*NONE

Does version record exist for the object in the from library?

- |     |                                                                                                                                                                                                                                                                                                                                                        |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Yes | Use this as the base for incrementing the version number based on application. How do we determine the "from object"? <ul style="list-style-type: none"> <li>▪ Line record: If the fields on the line are filled in, they are used.</li> <li>▪ Application: If the fields are empty, we will get the values from the application type code.</li> </ul> |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Find the next level of the application, starting with the target level, that the object is checked out from.

Does version record exist for the object in the checkout library?

- |     |                                                                        |
|-----|------------------------------------------------------------------------|
| Yes | Use this as base for incrementing version number based on application. |
|-----|------------------------------------------------------------------------|

No Starting at the level of the application where the checkout record was found, search up the application until you find a version record for the object, and use it as the base for incrementing the version number based on the application. If no record is found, then this is a new object and should be incremented from zero.

This extra bit of logic is to find objects that were checked out from perm-locked levels and the object does not exist yet in the highest unlocked level against which the checkout record will have been written.

### 3 If Version Increment Value is \*MAJOR

If the version increment value is \*MAJOR, we must determine the highest current major version number in the application and use it as the value we increment. To do this, we start at the lowest level of the application and work our way up the entire application structure, recording the highest major version level we find along the way. Whatever this value is will become the base value for incrementing.

This logic is necessary to catch the situation where an object in production has been recompiled. This change would have caused the major version number to be incremented since the object was checked out. If we didn't have this logic in the process, then the major version number could theoretically be reduced as a result of the promotion. This logic ensures that whenever the major version number is incremented, it is set to the highest value in the application.

---

## Examples of object version numbering

These examples are intended only to give you an idea for the dynamics at work.

### Example 1

#### 2-level App, Level 1 setting: \*MINOR, Level 2 setting: \*MAJOR)

Version 5.0 of object X is checked out from production. The object is changed and promoted to test. At the time of the promotion, the minor version number of this object is incremented in test and is now version 5.1. If the object were to be changed again before being promoted to production, the minor version number would continue to be incremented (5.2, 5.3...). When the object is promoted back to production, the version would be changed to 6.0.

### Example 2

#### 2-level App, Level 1 setting: \*MINOR, Level 2 setting: \*NONE

You ran the TSETAPPOBJ command to set the major version number for all objects within the application to 5.0. Whenever an object is modified, you want to increment the minor version number until you decide to make a new release. At that time, you would run the same procedure to set the major version number to 6.0. You could reverse the \*NONE and the \*MINOR parameters if you only wanted to increment when we promote to production.

### Example 3

**3-level App, Level 1 setting: \*MINOR, Level 2 setting: \*NONE, Level 3 setting: \*MAJOR**

Similar to example 1, here we define multiple test levels. This example shows how you would keep the version number from incrementing at each promotion. Presumably, you would only want to change the version number when the object was actually changed, rather than when it was promoted. If you incremented the minor version number with every promotion, this would mislead someone into thinking the object was revised twice before being promoted, when, in fact, it was only revised once.

However, one thing to consider is that the files that the programs are compiled over could be different at each level, so the underlying object would, in fact, be different. In that case, you might want to increment the minor version number at each level.

---

### Source-only types

TURNOVER® Lifecycle Manager tracks the same version information for source-only types, such as copybook code, but no object stamping is done.

---

### “Special” object types

TURNOVER® Lifecycle Manager tracks version information for special types, such as \*MSG IDs, but no object stamping is done.

---

## Distributing multiple versions

TURNOVER® Lifecycle Manager manages the distribution to several related applications so that all of the various objects and their modified versions arrive and are installed on the target computers correctly.

Forms for related applications are made dependent upon their base application forms. If they are run as a form group, they are distributed and installed in the correct order.

If you distribute the forms manually, the forms are still dependent, but you would have to be sure that both forms run to completion on the target computer.

Use extra care if you are distributing and promoting base and related application objects into the same target library. For example, you would not want to install local modifications and base objects on top of the modifications you made.

## 12 Cross-reference system

This section describes how TURNOVER® Lifecycle Manager's own cross-reference system works. If you have Hawkeye's Pathfinder or HelpSystems' Abstract cross-referencing tool and require setup guidance, please contact your Technical Support Representative as outlined on the SoftLanding website.

### Setting up for cross-references

To use TURNOVER® Lifecycle Manager's cross-reference, you must set the application definition parameter **X-Ref Method** to \*TURNOVER for *all application levels* that you want included in the cross-reference database.

When specifying the name of the **X-Ref Table** that will contain the cross-reference table, consider whether or not your applications intersect:

- If an application shares any programs or files with another application, they should share a cross-reference table.
- If they do not share any programs or files, they should have separate cross-reference tables.

Using a single cross-reference table ensures that you are aware of the true impact of a change across application boundaries. For example, a change to a file shared by AP, AR, and Purchasing programs will be evident when you make the file change.

### Updating the cross-reference database

Once you have defined the applications, run option 20 on the Work with Application Definition panel to submit a batch job that updates the cross-reference database. You must do this for each application level you want in the cross-reference.

Once the cross-reference database is built, you can update the database each time a form runs, or periodically if you set up a job to run forms on a regular basis. Set the application definition parameter **Update X-Ref on form run** to \*YES to have TURNOVER® Lifecycle Manager update the cross-reference during the Finalization phase of every form run.

### Accessing cross-reference usage information

To access cross-reference usage information, do one of these:

- Select option 15 (Check X-Ref) on the Programmer Worklist Manager

- Select option 20 (Work with object references) on the Checkout/Check-in Menu.
- Run the `WRKOBJREF` command.

## Cross-reference file layout

The primary cross-reference database consists of a single physical file located in the TURNOVER® Lifecycle Manager data library. This file consists of multiple members that you can use to separate cross-reference information among applications. The cross-reference application definition identifies the member as an *x-ref table*. The commands, `ADDXRFTBL` and `RMVXRFTBL`, are used to create or delete the x-ref table in the cross-reference file, along with all related logical files. All commands that maintain, view and work with cross-reference data have an **X-Ref table** parameter for identifying the x-ref table. TURNOVER® Lifecycle Manager manages the table using the **X-Ref Table** parameter value you supply in the application definition.

### TXRFDTAF file

Format: TXRFDTA

| Field name | Key | Type  | Length | Description           |
|------------|-----|-------|--------|-----------------------|
| TXOBJ      | Y   | *CHAR | 10     | Object                |
| TXLIB      | Y   | *CHAR | 10     | Library               |
| TXTYPE     | Y   | *CHAR | 10     | Object Type           |
| TXATTR     |     | *CHAR | 10     | Attribute             |
| TXROBJ     |     | *CHAR | 10     | Reference object      |
| TXRLIB     |     | *CHAR | 10     | Reference library     |
| TXRTYP     |     | *CHAR | 10     | Reference object type |
| TXRATR     |     | *CHAR | 10     | Reference attribute   |
| TXRMBR     |     | *CHAR | 10     | Reference member      |
| TXMTHD     |     | *CHAR | 1      | X-Ref method          |
| TXUSG      |     | *CHAR | 2      | Usage                 |
| TXADBY     |     | *CHAR | 10     | Added by user         |
| TXADDT     |     | *DEC  | 6.0    | Date added            |
| TXADTM     |     | *DEC  | 6.0    | Time added            |

## Fields

The first four fields in the file describe the object that uses the object identified in the next five fields.

Using the typical program-to-file relationship as an example, the program is identified by the information in the first four fields, and the file is identified by the information in the next five fields. There is one record for each file the program uses.

## X-Ref method

Defines the type of cross-reference information in the record. This is useful in both deleting and retrieving certain types of records from the file:

| Value | Description                             |
|-------|-----------------------------------------|
| 1     | Object (*PGM, *SRVPGM, *MODULE) to file |
| 2     | Object to object (program calls)        |
| 3     | Program (*PGM, *SRVPGM) to module       |
| 4     | Object to copybook                      |
| 5     | PF to LF                                |
| 6     | File to another file (Reference file)   |
| 7     | CMD to CPP, or VLDCKR                   |
| U     | User added/maintained                   |

## Usage

A 1-character code that describes the relationship between the objects.

A corresponding table keyed by the 1-character code contains an 8-character short description and 30-character long description to further identify the usage. The table contains several fixed values to be determined when the application is created, but the existence of the table allows you to make use of the user-defined portion of the cross-reference by adding new codes that would then be fully supported on all panels derived from the database. Some examples of codes that you might want in the file are:

Input, Output, Update, In/Out, PGM-CALL, CMD-CPP, CMD-VLD, COPY, and so on

Another field in the usage description table indicates whether the cross-reference item must be added to a form when adding the related item to a form. For example, when you add a file to a form, you might want all of the usage type 1 objects (such as programs) added to the form automatically. You may not want user-defined usage "U" types to be added automatically.

When adding lines to a form, TURNOVER® Lifecycle Manager checks the cross-reference to see if any related objects must also be added. If a form is large, it is possible to approach or exceed the maximum number of lines allowed on a form. To reserve space for the objects added during the cross-reference check, you can set an overflow threshold in the XREFOVRFLW data area. This threshold controls the number of lines that TURNOVER® Lifecycle Manager can add to a form before it begins adding cross-reference items to the form. The default number of lines that TURNOVER® Lifecycle Manager can add to a form before cross-referencing is 900.

The individuals creating the application can come up with the codes themselves to best suit the design of the application they create. The DSPPGMREF command, for example, produces output that has a 1-character usage field in it. If using this command to get cross-reference information, you would probably want to make use of the same codes used by that command, as a start, and then add additional codes as needed.

### Added by user, Date added, Time added

These three fields are for record keeping, especially as they apply to user-added cross-reference entries. They contain the user profile, date and time for when the record is added.

### Example of an x-ref table

The following table shows how the data might look in the file for the various types of information (not all fields are shown).

| TXOBJ    | TXLIB     | TXTYPE  | TXATTR | TXROBJ   | TXRLIB     | TXRTYP  | TXRATR | TXRMBR   | TXMTHD |
|----------|-----------|---------|--------|----------|------------|---------|--------|----------|--------|
| TURRPGMA | SOFTTURN  | *MODULE | RPG    | TAPPLF   | *LIBL      | *FILE   | PF     |          | 1      |
| TURRPGMA | SOFTTURN  | *MODULE | RPG    | TDFTSRCL | *LIBL      | *FILE   | LF     |          | 1      |
| TURRPGMA | SOFTTURN  | *MODULE | RPG    | TURCPGMB | *LIBL      | *PGM    | CLP    |          | 2      |
| TURRPGMA | SOFTTURN  | *MODULE | RPG    | QRPGSRC  | SOFTTURN31 | *FILE   | RPG    | TURRCPYA | 4      |
| TURRPGMX | SOFTTURN  | *PGM    | RPG    | TURRPGMA | SOFTTURN   | *MODULE | RPG    |          | 3      |
| TURRPGMX | SOFTTURN  | *PGM    | RPG    | TURCPGMB | SOFTTURN   | *MODULE | CLP    |          | 3      |
| TURSDSPA | SOFTTURN  | *FILE   | DSPF   | TURHDSPA | SOFTTURNE  | *PNLGRP | PNLGRP |          | U      |
| TUSROPTF | SOFTTURND | *FILE   | PF     | TUSROPTL | SOFTTURND  | *FILE   | LF     |          | 5      |
| TUSROPTF | SOFTTURND | *FILE   | PF     | TURNREFF | SOFTTURND  | *FILE   | PF     |          | 6      |
| WRKPRJ   | SOFTTURNE | *CMD    | CMD    | WRKPRJ   | *LIBL      | *PGM    | CLP    |          | 7      |
| WRKPRJ   | SOFTTURNE | *CMD    | CMD    | WRKPRJX  | *LIBL      | *PGM    | CLP    |          | 7      |
| WRKPRJ   | SOFTTURNE | *CMD    | CMD    | WRKPRJ   | SOFTTURNE  | *PNLGRP | PNLGRP |          | U      |

The information is derived in a variety of ways.

### Cross-referencing for \*SRVPGM objects

By default, TURNOVER® Lifecycle Manager includes \*SRVPGM objects in the cross-reference check it performs when creating a form. You can exclude these objects from TURNOVER® Lifecycle Manager's cross-reference check. To exclude the \*SRVPGM objects, create the SRVPGMXREF data area in the TURNOVER® Lifecycle Manager Data library and give it these attributes: a \*CHAR type, a length of 4, and a value of \*NO.

## Cross-referencing for C header files

In its cross-reference support for C header files, TURNOVER® Lifecycle Manager relates the files to their source objects (\*MODULE or \*PGM). This cross-referencing occurs automatically once you update your cross-reference database.

## User-defined relationships

APIs and exits enable to maintain this information automatically.

---

## Cross-reference commands

TURNOVER® Lifecycle Manager uses the following commands to maintain the cross-reference database. The IBM i object authority on the commands controls authority for all commands. Users QSECOFR and TURNOVER have authority to the commands. \*PUBLIC is \*EXCLUDE.

### ADDUSRREF: Add user reference

Adds a user-defined cross-reference record for a particular object.

### ADDXRFTBL: Add x-ref table

Adds a member with the same name as specified in the application definition **X-Ref Table** parameter, and then adds members of the same name to all logical files built over the physical file.

### LODOBJREF: Load object references

Loads the cross-reference data into the **X-Ref Table** for the objects specified. It handles loading data for a single object in a library or for all objects in a library. When you build cross-reference information for an application, TURNOVER® Lifecycle Manager runs this command once for each target library in the application level you selected, specifying \*ALL for object name. When a form runs, TURNOVER® Lifecycle Manager runs this command for each line of the form, specifying just the object on that line of the form.

Before loading the cross-reference information for an object, TURNOVER® Lifecycle Manager first removes information for that object. It deletes records where the object/library/type key fields match, and the usage field is not equal to 'U'. Maintenance of all user-defined cross-reference information is the responsibility of the user. Immediately after adding the information for an object, a user-exit program is called and passed the same parameters passed to the LODOBJREF command (except if \*ALL was specified for object, in which case the exit is called once for each object, passing the actual object name.)

The user program that loads the cross-reference information must build all types of the information that could exist for that object.



| Parameter | Description                                                                                                                                                                                                                                                                                                                                                        |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| OBJ       | The name of the object to be cross-referenced, or *ALL.                                                                                                                                                                                                                                                                                                            |
| LIB       | The name of the library that contains the object. *LIBL and *CURLIB are allowed, but not when *ALL is specified for object name.                                                                                                                                                                                                                                   |
| TYPE      | The object type of the object being cross-referenced. TYPE (*ALL) is allowed only in when *ALL is specified for object name. All other object types are supported, including the TURNOVER® Lifecycle Manager types defined in the TURNOVER® Lifecycle Manager type code file such as *MSGID. A user exit program to build information for that type can use these. |
| XRFTBL    | The name of the cross-reference table into which to load the information. This is a 10-character *NAME type field.                                                                                                                                                                                                                                                 |

### RMVOBJREF: Remove object references

Removes all non-user usage types from the cross-reference for a particular object. TURNOVER® Lifecycle Manager uses this command to remove cross-reference information from the database whenever an object is deleted by running a form, whether it be in production or in a test library. A user-exit program is called for each object for which information is removed.

This command removes object reference records where field TXOBJ matches the object specified. It does not remove records where the object was referenced. For example, if you remove information for PGMA, the command removes information for all of the files used by PGMA, programs called by PGMA, copybooks used, and so on. It does not remove information for programs that called PGMA, or commands that used PGMA as a processing program or validity checker.

| Parameter | Description                                                                                                                                                                                                                                                                                       |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| OBJ       | The name of the object to be removed, or *ALL.                                                                                                                                                                                                                                                    |
| LIB       | The name of the library that contains the object. *LIBL and *CURLIB are not be allowed in conjunction with object parameter of *ALL.                                                                                                                                                              |
| TYPE      | The object type of the object to be removed. TYPE (*ALL) should be allowed only in conjunction with *ALL for object name. Otherwise, all other object types are supported, including the all TURNOVER® Lifecycle Manager types that exist only within TURNOVER® Lifecycle Manager such as *MSGID. |
| XRFTBL    | The name of the cross-reference table containing the cross-reference information.                                                                                                                                                                                                                 |

### RMVUSRREF: Remove user reference

Removes all user-defined cross-reference records for a particular object.

## RMVXRFTBL: Remove x-ref table

Removes specified members from the cross-reference files.

## WRKOBJREF: Work with object references

Displays an object's usage in TURNOVER® Lifecycle Manager's cross-reference files. For more information, see *'Working with object references' on page 190*.

## WRKUSRREF: Work with user references

Enables you to define relationships between objects. For more information, see *'Working with user-defined references' on page 192*.

# Working with object references

Use the WRKOBJREF command to see all of the dependents of an object, or you can select one of the panel options that run this command. This command is also available as an option on the Checkout/Check-in menu.

- 1 On a command line, type WRKOBJREF, and then press Enter.

To run the command from a Programmer's Worklist or Synchronizer's Work with Change Impact File, you would select the corresponding option to add related items to your worklist or impact file.

| Work with Object References (WRKOBJREF)                             |                                      |
|---------------------------------------------------------------------|--------------------------------------|
| Type choices, press Enter.                                          |                                      |
| Object Name . . . . .                                               | _____ Name                           |
| Library Name . . . . .                                              | _____ Name                           |
| Type code . . . . .                                                 | _____ AAUDIT, ABPGM, ADMDL, ADPGM... |
| X-Ref Table Name . . . . .                                          | _____ APDEMO, DEFAULT, J52XREF, L... |
| Output . . . . .                                                    | * _____ *, *PRINT                    |
| <b>Bottom</b>                                                       |                                      |
| F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display |                                      |
| F24=More keys                                                       |                                      |

- 2 Type an object name, the name of the library where the object resides, the object's type, the cross-reference table name, and specify if you want the output displayed at a display station (\*) or printed (\*PRINT).

**Work with Object References (WRKOBJREF)**

Type choices, press Enter.

|                            |           |                                |
|----------------------------|-----------|--------------------------------|
| Object Name . . . . .      | TLINEF    | Name                           |
| Library Name . . . . .     | SOFTTURND | Name                           |
| Type code . . . . .        | PF        | AAUDIT, ABPGM, ADMDL, ADPGM... |
| X-Ref Table Name . . . . . | DEFAULT   | APDEMO, DEFAULT, J52XREF, L..  |
| Output . . . . .           | *         | *, *PRINT                      |

**Bottom**

F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display  
F24=More keys

### 3 Press Enter.

MM/DD/YY 17:25:15 **Work with Cross-Reference** Your Company, Inc.  
X-Ref Depth= 01

Worklist . . . . APPR004 **Chg credit-term verify program**  
Data set . . . . DEFAULT Apply filters . . . Y (F17=Filters)

1=Add to Worklist 2=Add to Worklist/Checkout 4=Compile in Development library  
5=View checkout 6=Work with relationships 8=Browse member

----- Uses/Used-by ----->

| Object | Library   | Attr | Object     | Library   | Attr | How Used  | Out | U/B | Chk |
|--------|-----------|------|------------|-----------|------|-----------|-----|-----|-----|
| TLINEF | SOFTTURND | PF   | ADDCHKITMR | SOFTTURN  | RPG  | Input     | N   | B   |     |
| TLINEF | SOFTTURND | PF   | ADDCMDR    | SOFTTURN  | RPG  | Input     | N   | B   |     |
| TLINEF | SOFTTURND | PF   | ADDCMDXR   | SOFTTURN  | RPG  | Input     | N   | B   |     |
| TLINEF | SOFTTURND | PF   | CHGFORMPRR | SOFTTURN  | RPG  | Update    | N   | B   |     |
| TLINEF | SOFTTURND | PF   | CVTLINCMTR | SOFTTURN  | RPG  | Input     | N   | B   |     |
| TLINEF | SOFTTURND | PF   | DISTDATAR  | SOFTTRUN  | RPG  | Input     | N   | B   |     |
| TLINEF | SOFTTURND | PF   | TCRTPGM    | SOFTTURN  | CLP  | Not given | N   | B   |     |
| TLINEF | SOFTTURND | PF   | TCRTSRVPGM | SOFTTURN  | CLP  | Not given | N   | B   |     |
| TLINEF | SOFTTURND | PF   | TLINEH     | SOFTTURND | LF   | PF->LF    | N   | B   |     |
| TLINEF | SOFTTURND | PF   | TLINEL     | SOFTTURND | LF   | PF->LF    | N   | B   |     |
| TLINEF | SOFTTURND | PF   | TLINEN     | SOFTTRUND | LF   | PF->LF    | N   | B   |     |

**More...**

F3=Exit F4=Prompt F5=Refresh F10=Data set information F12=Cancel  
F13=Repeat F17=Filters F19=Submit 1/2 F21=Command line F22=Print

You can also access this panel from your worklist (option 15=Check X-Ref), and look at objects related to one or more items at once. For more information, see 'Checking for Cross-reference information' in the *TURNOVER® Lifecycle Manager User Guide*.

- The **U/B** column indicates whether the object you are researching (listed at the left) uses (U) or is used by (B) the reference object.
- The **How Used** column tells you how the object uses or is used by the reference object.

### 4 Press F17 to subset the list of objects you see on the Work with Cross-Reference panel. For example, you may want to see only the reference objects (in this case, programs) that update (UPD) the file you are researching.

```

MM/DD/YY 17:25:15          Work with Cross-Reference          Your Company, Inc.
.....
:
: Type filter values, press Enter.
:
: Attribute . . . . . _____ F4=List
: Library . . . . . _____ Library name
: How used . . . . . _____ INP, UPD, OUT
: Checked out . . . . . Y, N, blank
: X-Ref type . . . . . B=Used by, U=Uses
:
:
:
:
: F3=Exit   F4=List   F12=Cancel
:
:
:
:.....
      TLINEF      SOFTTURND  PF      TLINE      SOFTTURND  LF      PF->LF      N  B
                                           More...
F3=Exit  F4=Prompt  F5=Refresh  F10=Data set information  F12=Cancel
F13=Repeat  F17=Filters  F19=Submit 1/2  F21=Command line  F22=Print
  
```

5 Press F10 to see a list of libraries over which the cross-reference is built.

```

MM/DD/YY 17:26:35          Work with X-Ref Data Set Information  Your Company, Inc.

Data set . . . . .TURN40

Library  Appl  Rel  Ver  Lev  Last Appl  Last form
          X-Ref Update  X-Ref Update
SOFTTURN  TO    3    2    2    4/17/17    3/15/16
SOFTTURND TO    3    2    2    4/17/17    3/15/16
SOFTTURN  TO    3    2    2    4/17/17    3/15/16
SOFTTURN4 TO    4    0    2    4/17/17    4/17/17
SOFTTURN4D TO   4    0    2    4/17/17    4/17/17
SOFTTURN4E TO   4    0    2    4/17/17    4/17/17

                                           Bottom
F3=Exit   F12=Cancel
  
```

This panel displays all of the libraries included in the cross-reference table (X-Ref table) and the applications associated with each one.

## Working with user-defined references

Use the `WRKUSRREF` command to add some of your own references to a cross-reference table.

1 On a command line, type `WRKUSRREF`, and then press Enter.

The Work with User References (WRKUSRREF) panel appears.

```

                                Work with User References (WRKUSRREF)

Type choices, press Enter.

X-Ref Table Name . . . . . _____ APDEMO, DEFAULT, J52XREF, L...

                                                                 Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
Parameter XRFTBL required.
    
```

- 2 Type the name of a cross-reference table, and then press Enter.

The Work with User Defined References panel appears.

- 3 Press F6 to add a cross-reference.

The Add User Defined Cross Reference panel appears.

```

MM/DD/YY 17:45:12      Work with User Defined References      Your Company, Inc.
                        Cross Reference Table: APDEMO           YOURSYS

2=Chan                  Add User Defined Cross Reference      --

  Obje  Object. . . . . _____
        Library . . . . . _____
        Object Type . . . . . _____
        Attribute . . . . . _____
        Reference Object. . . . . _____
        Reference Library . . . . . _____
        Reference Object Type . . . . . _____
        Reference Attribute . . . . . _____
        Reference Member. . . . . _____
        Usage . . . . . _      F4-List

        F12=Cancel

F3=Exit  F6=Add X-Ref  F12=Cancel  F21=System command
    
```

- 4 Type an object name, the name of the library where the object is, the object's type, and the object's TURNOVER® Lifecycle Manager type code (attribute).
- 5 Type the same information for the existing object to which you are relating the new object.
- 6 Specify how the new object is using the referenced object. Press F4 to display a list of values from which you can choose.

```

MM/DD/YY 17:45:12      Work with User Defined References      Your Company, Inc.
                        Cross Reference Table: APDEMO           YOURSYS

2=Chan                  Add User Defined Cross Reference      --

  Obje  Object. . . . . JIMPGM
        Library . . . . . BROWN
        Object Type . . . . . *PGM
        Attribute . . . . . RPG
        Reference Object. . . . . APPF001
        Reference Library . . . . . PRODTST
        Reference Object Type . . . . . *FILE
        Reference Attribute . . . . . PF
        Reference Member. . . . .
        Usage . . . . . A4 PGM->FILE      F4-List

        F12=Cancel

F3=Exit  F6=Add X-Ref  F12=Cancel  F21=System command
  
```

7 Press Enter.

TURNOVER® Lifecycle Manager displays "Record added".

8 Press F12.

The Work with User Defined References panel displays the reference (in alphabetical order by object name) in the list of references that are defined for the cross-reference table:

```

MM/DD/YY 17:45:12      Work with User Defined References      Your Company, Inc.
                        Cross Reference Table: APDEMO           YOURSYS

2=Change  4=Delete

                        Reference -----
  Object  Library  Type  Attr  Object  Library  Type  Attr
  _  ACMPGM  BROWN  *PGM  RPG  APPF001  PRODTST  *FILE  PF
  _  JPMPLM  BROWN  *PGM  RPG  APPF001  PRODTST  *FILE  PF
  _  JOPGM   BROWN  *PGM  RPG  APPF001  PRODTST  *FILE  PF

                        Bottom

F3=Exit  F6=Add X-Ref  F12=Cancel  F21=System command
  
```

9 Use option 2 to change any reference listed or option 4 to delete a reference.

# 13 Promoting objects into independent ASPs

Independent ASPs, or “iASPs,” allow disk units to be taken offline or placed online independently of the system.

iASPs are different from regular ASPs, which are numbered 1 – 32 and cannot contain libraries with names that exist in other areas of the system. iASPs are referred to by their group name, and they can contain libraries with names that exist in other system areas. The `SETASPGRP` (Set ASP Group) command tells the system *which* library to actually look at.

## Requirements

You can manage objects in iASPs, as long as you understand and work within these **TURNOVER®** Lifecycle Manager limitations:

- It is your responsibility to ensure the proper ASP group has been set prior to working in this environment. This is either by running the `SETASPGRP` command or by using the `INLASGRP` parameter on the job description.
- You cannot promote objects from one ASP group to another. The ASP group should be set at the beginning of a job, ideally on the job description, and not changed during the job.
- You cannot explode from one ASP group to another. Therefore you cannot explode to the same library in different ASP groups.
- All libraries specified on a form run (library list, target libraries, from libraries, Job description library, and so on) must be in the same ASP group or in user ASPs, accessible to the form run.
- The ASP group should be set prior to creating a recovery form or error checking any form.
- If receiving forms, the receive job may need to have the correct initial ASP group in order to receive correctly.
- Using `SETASPGRP` in a pre-run command, post-run command, or exit point could adversely affect the library list of the running form and should be avoided. The ASP group should ideally be set using the job description defined in the application for the form. The next best solution would be to use the form pre-run exit.
- **TURNOVER®** Lifecycle Manager cross-reference does not recognize if related objects exist in different ASP groups.

- Object History, Checkout History, and object auditing will not tell you which ASP group an object belongs to. It treats it just as a specific library. Objects that exist in the same library in different ASP groups will all look the same to TURNOVER® Lifecycle Manager's history and especially its auditing. Changed outside of TURNOVER® Lifecycle Manager messages will likely be the norm in this environment.
- You cannot archive offline forms that were created for different ASP groups.
- If exploding objects to 485 or more libraries, the base object cannot be in ASPs 17 – 32.
- TURNOVER® Lifecycle Manager libraries can be installed into an iASP. See Getting Started with TURNOVER® Lifecycle Manager installation guide for more information.

## Setting iASP in job descriptions

The following jobs have used the INLSPGRP(\*JOBID) parameter when they are submitted:

- Form Run (*Tform#G*)
- Form Finalization (*Tform#F*)
- Form Receive (*TOAUTORCV*)

The job description to be used is the job description specified in the form header. The exception to this is the form receive process, which submits jobs using the TURNOVER job description.

**Note** Other processes in TURNOVER® Lifecycle Manager that submit jobs will not explicitly specify the INLSPGRP parameter, so the default value of INLSPGRP(\*CURRENT) will be used. This means that you must take care to set the ASP group yourself, either by specifying INLSPGRP in your own job description, or by using the SETASPGRP command from a command line.

## TURNOVER® Lifecycle Manager defaults for iASP promotions

When promoting objects into iASPs, TURNOVER® Lifecycle Manager uses these default library names and settings:

- Archive libraries for ASPs 17 - 32 are named *Tform#DH* through *Tform#DW*.
- Archive libraries for iASPs are named *Rform#xx*, where *R* is resolved from the 1-character data area TARCHPRE, and *xx* is a base 36 sequential number.

The value shipped in TARCHPRE is R, but you can change this to another letter (except for A, Q, T, W, X, or Z).



# 14 Managing IFS objects

IFS stands for *integrated file system*. A *file system* is the way a computer's operating system stores data in logical units called files, directories, libraries, and objects. The IBM i operating system supports several different file systems, and includes a *common integrated file system interface* for accessing data stored in any of the supported file systems.

TURNOVER® Lifecycle Manager enables you to manage objects stored in any of the IBM i-supported file systems including IBM i. You can manage technologies such as:

- JAVA, HTML, and other internet-related applications
- PC-based word or graphic processing applications
- client/server development tools.

---

## Summary of TURNOVER® Lifecycle Manager changes for IFS

IFS support in TURNOVER® Lifecycle Manager is API-driven. To use it, you must change some the following information.

### Global defaults

See '[Global defaults for IFS](#)' on page 199.

### Global type codes

See '[Global type codes for IFS](#)' on page 201.

### Application definitions

- IFS objects require Path defaults instead of Library defaults. To specify these, press F8 on the Library Defaults panel.
- Form line requirements are different for IFS objects. To specify form line default information for IFS objects, press F8 on the Line Defaults panel.
- Include IFS type codes in your application's type code list.

For more information about the application definition requirements for IFS, see *'Application definitions' on page 197*.

## Commands and APIs

For information about the commands for checking IFS objects in and out, for adding them to forms, and for linking commands to IFS form lines, see *'Managing IFS objects' on page 207*.

## IFS object information

- On the **Object History** panels, you can display for IFS object information, including checkout history, archive history and management, and distribution history. For more information, see *'IFS object history' on page 215*.
- On the **Check out/Check in source** menu, you can view checked out IFS objects. For more information, see *'Checking out IFS objects' on page 208*.

## IFS exit points

Exit points for custom processing at various stages of IFS object management are available. They include pre- and post-checkout exits, pre- and post-line exits, and a check-in exit. For more information, see *'IFS exit points' on page 220*.

---

## IFS object authority

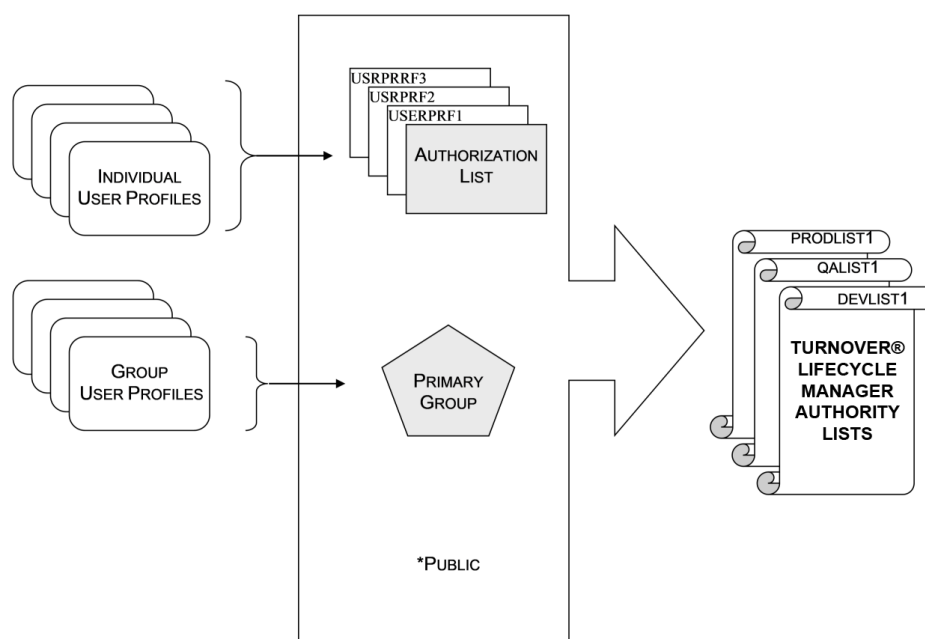
During a TURNOVER® Lifecycle Manager form run, TURNOVER® Lifecycle Manager adopts ownership of objects in order to create them in new places and set authority parameters for them. However, this concept does not apply to IFS commands. Instead, you specify in your TURNOVER® Lifecycle Manager Global Defaults a special user profile that TURNOVER® Lifecycle Manager uses to run IFS commands (parameter **IFS Adopted authority user**). For more information, see *'IFS Adopted authority user' on page 200*.

When creating IBM i file system objects, TURNOVER® Lifecycle Manager can use existing objects known as *reference objects* to set authority parameters. This is because the `GRTOBJAUT` command's `REFOBJ` parameter enables you to specify an object to use as an authority model. IFS object authorities are assigned using a different command, called `CHGAUT`. It has no parameter for specifying reference objects.

Instead, TURNOVER® Lifecycle Manager applies IFS object authorities by using a user profile-based authority system. IBM i system entities known as *authorization lists* and *primary groups* control who has access to what file system directories. (Authorization lists and primary groups are two different ways of grouping user profiles for the purpose of assigning authority to objects.)

For these reasons, the IFS line defaults have a parameter for a TURNOVER® Lifecycle Manager *authority list*. The IFS authority list is a collection of data and object authority values that are assigned using primary groups, authorization lists, and the \*PUBLIC system authority designation. When combined into an authority list, these values control assignment of IFS object authority at each level of your application. You can create a different TURNOVER® Lifecycle Manager authority list for each level of your application, if necessary.

### TURNOVER® Lifecycle Manager authority lists



## Setting up TURNOVER® Lifecycle Manager to manage IFS objects

To use TURNOVER® Lifecycle Manager to manage IFS objects, you must configure:

- some TURNOVER® Lifecycle Manager global default values: see below for more information..
- some TURNOVER® Lifecycle Manager global type codes: see ['Global type codes for IFS' on page 201](#)
- some parts of any application definition you plan to use for managing IFS objects: see ['Application defaults and rules' on page 202](#).

### Global defaults for IFS

These global TURNOVER® Lifecycle Manager defaults apply to IFS:

- IFS Adopted authority user
- IFS archive subdirectory.

### To set the global defaults

On the main menu, select option 8 (Utility menu), and then option 5.

The Set Global TURNOVER Defaults panel appears.

Set Global TURNOVER Defaults (SETGLOBE)

Type choices, press Enter.

Manage S/36 object types . . . . .

> N

\*SAME, Y, N

Allow group checkout . . . . .

> N

\*SAME, Y, N

Use Delete Test Object rule . . .

> Y

\*SAME, Y, N

Company name . . . . .

> 'Your Company'

Dft pre/post run cmd lib list . .

> \*CURRENT

\*SAME, \*CURRENT, \*APPL

Specify method of distribution . .

> \*DISTRIB

\*SAME, \*FORM, \*DISTRIB

IFS Adopted authority user . . .

>

Name, \*SAME, \*CURRENT

IFS Archive subdirectory . . . .

> DEFAULT

Character value

Bottom

F3=Exit

F4=Prompt

F5=Refresh

F12=Cancel

F13=How to use this display

F24=More keys

### IFS Adopted authority user

This parameter controls whether or not TURNOVER® Lifecycle Manager adopts the authority of a specially configured user profile on your system during checkout processing and promotion.

| Value       | Description                                                                                                                                                                                                                            |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>name</i> | TURNOVER® Lifecycle Manager uses the named profile's authority when moving or creating objects and assigning object authorities.                                                                                                       |
|             | If you plan to have TURNOVER® Lifecycle Manager adopt authority, create a user profile specifically for the purpose.                                                                                                                   |
|             | The profile must have *ALLOBJ authority or be a member of *SECOFR user class, or have appropriate authority for the IFS objects that TURNOVER® Lifecycle Manager will manage.                                                          |
| *CURRENT    | TURNOVER® Lifecycle Manager uses the authority settings of the user associated with the job to move or create objects and assign authorities. The user must be able to manage object existence and authority, or processing will fail. |

### IFS Archive subdirectory

Specify a default subdirectory name for all archived IFS objects. You can override this value at the application level: see *'Application defaults and rules' on page 202*.

IFS objects are archived in subdirectories of the /TURNOVER/ARCHIVE subdirectory, in the form:

/TURNOVER/ARCHIVE/ssssssss/ffffff

where *sssssss* is a subdirectory name you can define with this global default, and *ffffff* is the form number associated with the object being archived.

## Global type codes for IFS

Global type codes for managing JAVA, CLASS and HTML files are included with TURNOVER® Lifecycle Manager. You can copy these and modify them as necessary to define your own global IFS type codes.

### To set the global type codes

- 1 On the main menu, select option 8 (Utility menu), and then select option 4.

|                                                                         |                         |                               |
|-------------------------------------------------------------------------|-------------------------|-------------------------------|
| MM/DD/YY<br>08:50:54                                                    | Change an IFS Type Code | Your Company, Inc.<br>YOURSYS |
| Type code . . . . .                                                     | HTML                    |                               |
| Description . . . . .                                                   | HTML files              |                               |
| Explode . . . . .                                                       | <u>N</u>                |                               |
| Deploy . . . . .                                                        | <u>N</u>                |                               |
| Case sensitive . . . . .                                                | <u>Y</u>                |                               |
| Sequence . . . . .                                                      | <u>*AFTER</u>           |                               |
| F3=Exit F6=Add default linked command F7=Select Applications F12=Cancel |                         |                               |

- 2 Set the parameters that are described below, and then press Enter.

### Type code

Specify a unique identifier for the type code.

The code can include letters and numbers. The maximum length is 6 characters.

### Description

Describe what kind of object this type code will be used to manage.

### Explode

| Value | Description                                                                                                    |
|-------|----------------------------------------------------------------------------------------------------------------|
| Y     | Objects of this type can be copied, or “exploded,” into specific paths defined by your application definition. |
| N     | Objects of this type should not be exploded.                                                                   |

## Deploy

| Value | Description                                                                                                                        |
|-------|------------------------------------------------------------------------------------------------------------------------------------|
| Y     | Objects of this type should be copied to workstation deployment paths specified in the application's IFS explode/deployment table. |
| N     | Objects of this type should not be deployed.                                                                                       |

## Case sensitive

If the file system to which objects of this type will be promoted is case-sensitive (for example, QOpenSys), specify Y.

If the file system is not case-sensitive, specify N.

## Sequence

The sequence for IFS type codes has two parts: IFS object sequence and form object sequence. The three-digit number parameter controls how IFS objects are processed in relation to each other.

| Value   | Description                                                       |
|---------|-------------------------------------------------------------------|
| *BEFORE | IFS objects are processed before any non-IFS objects on the form. |
| *AFTER  | IFS objects are processed after any non-IFS objects on the form.  |

## Linking type codes to commands

Use F6 to link to this type code a default command that will always be run when promoting objects of this type.

For example, if you are promoting a JAVA class object that runs on an IBM i web server, you can optimize it by attaching the IBM i CRTJVAPGM command to the type code you will use to promote the object. (A CLASS type code is shipped pre-configured for this capability.) For more information, see *'Optimizing Java programs' on page 223*.

---

# Application defaults and rules

To manage IFS objects, your TURNOVER® Lifecycle Manager application needs to know where to archive them; where to get them from and where to put them at each application level; how to assign authority to newly created objects at each level; and how to handle them at promotion time.

If your application includes IFS type codes, an extra field called **IFS Archive subdirectory** appears on the Application Defaults and Rules panel.

|                                                                        |                                                                         |                                        |
|------------------------------------------------------------------------|-------------------------------------------------------------------------|----------------------------------------|
| MM/DD/YY<br>14:34:53                                                   | Application Defaults and Rules<br>Application: DOC Rel: 5 Ver: 1 Lev: 1 | Your Company<br>SYSTEM: YOURSYS        |
| Type choices, press Enter.                                             |                                                                         | Field attributes<br>R=Restrict, L=Lock |
| Source archive file . . . . . ARCHIVE                                  |                                                                         |                                        |
| Library . . . . . *DEFAULT                                             |                                                                         |                                        |
| Archive objects . . . . . Y Y,N L                                      |                                                                         |                                        |
| IFS Archive subdirectory . . . . . *DEFAULT                            |                                                                         |                                        |
| Field mapping default for copy file command . . . . . *MAP *DROP       |                                                                         |                                        |
| Compress out deleted records on copy file command . . . *YES *YES, *NO |                                                                         |                                        |
| Require source checkout . . . . . Y Y,N                                |                                                                         |                                        |
| Checkout error severity . . . . . E W,E,blank                          |                                                                         |                                        |
| X-Ref method . . . . . *NONE *TURNOVER, *HAWKEYE, *ABSTRACT, *NONE     |                                                                         |                                        |
| F3=Exit F12=Cancel                                                     |                                                                         |                                        |

### IFS Archive subdirectory

You can override the TURNOVER® Lifecycle Manager Global Default by specifying a subdirectory name where TURNOVER® Lifecycle Manager should archive IFS objects managed by this application.

To use the global default value, type \*DEFAULT.

## IFS path defaults

IFS objects are stored in directories, instead of libraries. You must specify the subdirectories where your IFS objects are.

### To specify the subdirectories

On the Library Defaults panel, press F8.

The Path Defaults panel appears.

| MM/DD/YY                                                  | Path Defaults                         | Your Company |
|-----------------------------------------------------------|---------------------------------------|--------------|
| 17:01:38                                                  | Application: DOC Rel: 5 Ver: 1 Lev: 1 | SYSTEM: SLS  |
| Type choices, press Enter.                                |                                       |              |
| Target path . . . /Docs/Turnover/Review                   |                                       |              |
| _____ F22=Full path                                       |                                       | Locked: N    |
| From path . . . . /Docs/Turnover/Work                     |                                       |              |
| _____ F22=Full path                                       |                                       | Locked: N    |
| F3=Exit F12=Cancel F19=Explode/deploy paths F22=Full path |                                       |              |

Separate directory names by using / (slashmark).

To “explode” IFS objects into other directories, press F19. Explode paths for IBM i IFS objects can have a different authority list than the target promotion path.

## IFS line defaults

The IFS Line Defaults panel tells TURNOVER® Lifecycle Manager how to create the new IFS object in the target directory (copy or move), and whether to delete the old object during the course of the form promotion. (See *‘IFS object authority’ on page 198.*)

| MM/DD/YY                                    | IFS Line Defaults                     | Your Company    |
|---------------------------------------------|---------------------------------------|-----------------|
| 16:36:53                                    | Application: DOC Rel: 5 Ver: 1 Lev: 1 | SYSTEM: SLS     |
| Type choices, press Enter.                  |                                       |                 |
| Promotion method . . . . .                  | <u>CPY</u>                            | CPY, MOV, *NULL |
| Locked . . . . .                            | <u>N</u>                              | Y, N            |
| Target authority list . . . . .             | <u>DOCUSE</u>                         | Name, F4=List   |
| Locked . . . . .                            | <u>N</u>                              | Y, N            |
| Delete from object . . . . .                | <u>Y</u>                              | Y, N            |
| Locked . . . . .                            | <u>N</u>                              | Y, N            |
| Development authority list . . . . .        | <u>DOCDEV</u>                         | Name, F4=List   |
| F3=Exit F4=Select authority list F12=Cancel |                                       |                 |



## Building TURNOVER® Lifecycle Manager authority lists

To control the assignment of authority to the objects it creates, TURNOVER® Lifecycle Manager uses an internal object called an *authority list*. (This is different from the IFS object called an *authorization list*. An authorization list is just one piece of a TURNOVER® Lifecycle Manager authority list.)

A TURNOVER® Lifecycle Manager authority list is a container in which you can collect:

- individual user profile information
- IFS authorization list information
- primary group information.

See also '*IFS object authority*' on page 198.

When you set up an application to use IFS, you must create one or more authority lists. For example, you might create an authority list that sets full access authorities for development, and one that limits access for the non-development levels of your application.

### To create an authority list

- 1 Move the cursor to one of the authority list fields, and then press F4.
- 2 Copy the SAMPLE authority list.
- 3 Specify a name for the authority list.

The name must not include spaces.

- 4 Use option 2 to edit the authority list.

The Change IFS Authority List Object panel appears.

| MM/DD/YY                               | Change IFS Authority List Object         |                            | Your Company |
|----------------------------------------|------------------------------------------|----------------------------|--------------|
| 18:17:40                               |                                          |                            |              |
| Type choices and press Enter.          |                                          |                            |              |
| Authority list object . . . . .        | MYNEWLIST                                |                            |              |
| Description . . . . .                  | Development level authority list for IFS |                            |              |
| Owner . . . . .                        | *USRPRF                                  | Name, *USRPRF              |              |
| Authorization list . . . . .           | *NONE                                    | Name, *NONE                |              |
| Data authorities . . . . .             | *R                                       | *NONE, *RWX, *RX, *RW,...  |              |
| Object authorities . . . . .           | *OBJREF                                  | *NONE, *ALL, *OBJEXIST,... |              |
| Primary group . . . . .                | *NONE                                    | Name, *NONE                |              |
| Primary group data authorities . . . . | *RWX                                     | *NONE, *RWX, *RX, *RW,...  |              |
| Primary group object authorities . . . | *ALL                                     | *NONE, *ALL, *OBJEXIST,... |              |
| F3=Exit F12=Cancel                     |                                          |                            |              |

- 5 Set the parameters that are described below, and then press Enter.

### **Description**

Described what kind of authority this list assigns.

### **Owner**

Specify the owner of the objects that are managed by the application.

To assign ownership to the user who creates the object, type \*USRPRF.

### **Authorization list**

If an existing IFS authorization list assigns appropriate authority to the appropriate users, specify it here.

Otherwise, specify \*NONE, and then assign authorities by using the rest of the fields.

### **Object authorities**

Specifies \*PUBLIC ability to manage the existence of this application's IFS objects. For information about the values, see the online help.

### **Primary group**

Specify a group user profile as the primary group for the IFS objects associated with this application. A group user profile is created as a means of easing the task of assigning authorities on systems where there are large numbers of users.

### **Primary group data authorities**

Specify the data access authority to be granted to the users in the primary group when new objects are created for this application. For information about the values, see the online help.

### **Primary group object authorities**

Specifies the object authority to be granted to the users in the primary group when new objects are created for this application. For information about the values, see the online help.

---

## **IFS application type codes**

When you include a global IFS type code in your application definition, it becomes an application type code with its own application-based default values. The application type code is used to override (and, in some cases, lock) application default values from other areas of the application definition at the object level, giving you more control over what happens to different kinds of objects at promotion time.

## IFS application type code definition panel

|                                                           |                                       |                                |
|-----------------------------------------------------------|---------------------------------------|--------------------------------|
| MM/DD/YY                                                  | IFSDOC Defaults                       | Your Company, Inc.             |
| 14:49:40                                                  | Application: DOC Rel: 5 Ver: 1 Lev: 1 | SYSTEM: YOURSYS                |
| Type choices and press Enter.                             |                                       |                                |
| Target path . . . /DOCS/TURNOVER/REVIEW                   |                                       | F22=Full path Locked: <u>N</u> |
| From path . . . . /DOCS/TURNOVER/WORK                     |                                       | F22=Full path Locked: <u>N</u> |
| Promotion method . . . . . <u>CPY</u> CPY, MOV, *NULL     |                                       |                                |
| Locked . . . . . <u>N</u> Y, N                            |                                       |                                |
| Authority list . . . . .                                  |                                       | F4=List                        |
| Locked . . . . . <u>N</u> Y, N                            |                                       |                                |
| Delete from object . . . . . <u>Y</u> Y, N                |                                       |                                |
| Locked . . . . . <u>N</u> Y, N                            |                                       |                                |
| Archive . . . . . <u>Y</u> Y, N                           |                                       |                                |
| Locked . . . . . <u>L</u> Y, N                            |                                       |                                |
| Explode . . . . . <u>Y</u> Y, N                           |                                       | Deploy . . . <u>N</u> Y, N     |
| Distribute . . . . . <u>Y</u> Y, N                        |                                       |                                |
| Development authority list . . . . <u>DOCDEV</u>          |                                       | F4=List                        |
| F3=Exit F4=Select authority list F12=Cancel F22=Full path |                                       |                                |

For more information, use the online help.

---

## Managing IFS objects

**Note** These instructions assume that you are using an IBM i system to manage IFS objects. However, you are recommended to use the TURNOVER® Client application for managing PC-based files.

Managing IFS objects with TURNOVER® Lifecycle Manager is much the same as for any other IBM i object. The process consists of three main steps:

- 1 Check out the object: see *'Checking out IFS objects' on page 208*.
- 2 Change the object and verify the change.

Once your IFS objects are checked out, you can edit them in the target directory, using an editing program is appropriate for the kind of object you are working with. You cannot edit IFS objects from within TURNOVER® Lifecycle Manager.

You might also do some compiling, testing, or reviewing at this stage before promoting to a QA or production directory.

- 3 Promote the object: see *'Promoting IFS objects' on page 210*.

---

## Summary of commands for managing IFS objects

These TURNOVER® Lifecycle Manager commands support object management activities for IFS objects.

| Description                        | Usage                                                                                                            | API        |
|------------------------------------|------------------------------------------------------------------------------------------------------------------|------------|
| Checking out IFS objects           | Command line, program, or Checkout menu option 11<br><br><i>See 'Checking out IFS objects' on page 208.</i>      | TCHKOUTIFS |
| Checking in IFS objects            | Command line, program, or Checkout menu option 24<br><br><i>See 'Checking in IFS objects' on page 209.</i>       | TCHKINIFS  |
| Adding IFS objects to forms        | Program only (includes return parameters)                                                                        | TADDIFSLIN |
|                                    | Command line or program<br><br><i>See 'Adding IFS objects to a form' on page 211.</i>                            | TADDIFSLNI |
| Linking commands to IFS form lines | Program only (includes return parameters)                                                                        | TADDIFSCMD |
|                                    | Command line or program<br><br><i>See 'Linking pre-run and post-run commands to IFS form lines' on page 212.</i> | TADDIFSCMI |

## Checking out IFS objects

- To check out IFS objects, do one of the following:
  - On the main menu, select option 5. On the Check-out/Check-in Menu, select option 11.
  - On a command line, type `TCHKOUTIFS`.

```

Check Out IFS Object (TCHKOUTIFS)

Type choices, press Enter.

Object . . . . . _____

Type code . . . . . _____ CLASS, DLL, EXE, HLP, HTML, ...
Application . . . . . _____ Character value
Reference . . . . . _____ Character value

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
    
```

- 2 Specify the IFS object you want to check out, supplying information about where the object can be found, where to copy it, its object type, application and reference information, and so on.

To display additional parameters, press F10.

For more information about the parameters, see the online help.

- 3 Press Enter.

A checkout record is written to the IFS checkout file and the object is copied according to your instructions.

---

## Checking in IFS objects

Normally, check-in occurs automatically when you run a form.

You might check in objects manually only if you checked them out by accident, or if you realize that an object you checked out does not need to be changed.

- 1 To check in IFS objects, do one of the following:

- On the main menu, select option 5. On the Check-out/Check-in menu, select option 24.
- On a command line, type `TCHKINIFS`.

| MM/DD/YY                                                                                                                   | Check in IFS Objects | Your Company          |
|----------------------------------------------------------------------------------------------------------------------------|----------------------|-----------------------|
| Position to object:                                                                                                        |                      | Filter by Programmer: |
|                                                                                                                            |                      | Reference:            |
|                                                                                                                            |                      | Appl/Rel/Ver/Lev:     |
| 1=Checkin 2=Checkin, delete test object                                                                                    |                      |                       |
| Object                                                                                                                     | Type                 | Programmer Reference  |
| AUTOMARK.DOC                                                                                                               | IFSDOC               | MARYS PROJ0003        |
| Path: /DOCS/TURNOVER/PUBLISH/R51                                                                                           |                      |                       |
| Application: DOC Release: 05 Version: 01 Level: 02                                                                         |                      |                       |
| DISTRIB.DOC                                                                                                                | IFSDOC               | MARYS PROJ0003        |
| Path: /DOCS/TURNOVER/PUBLISH/R51                                                                                           |                      |                       |
| Application: DOC Release: 05 Version: 01 Level: 02                                                                         |                      |                       |
| EXITS.DOC                                                                                                                  | IFSDOC               | MARYS PROJ0003        |
| Path: /DOCS/TURNOVER/PUBLISH/R51                                                                                           |                      |                       |
| Application: DOC Release: 05 Version: 01 Level: 02                                                                         |                      |                       |
| FILES.DOC                                                                                                                  | IFSDOC               | MARYS PROJ0003        |
| Path: /DOCS/TURNOVER/PUBLISH/R51                                                                                           |                      |                       |
| Application: DOC Release: 05 Version: 01 Level: 02                                                                         |                      |                       |
| Note: Objects being promoted on a TURNOVER® Lifecycle Manager form are checked in automatically. Do not check in manually. |                      |                       |
| F3=Exit                                                                                                                    | F22=Full path/object |                       |

This panel shows a list of IFS objects that are checked out. You can limit what you see in the list by filtering for a specific programmer, reference, and application code, release, version and/or level.

If the **Path** is not completely visible, press F22.

- 2 Select the object you want to check back in, and then press Enter.

---

## Promoting IFS objects

Once you have changed the IFS objects, you can promote them to your testing directories, by using a TURNOVER® Lifecycle Manager form.

---

## Creating forms

### Using the interface

- 1 On the main menu, select option 6 to start Work with Forms, and then press F6 to add a form.
- 2 Specify a task reference and any other relevant information, and press Enter.

During this process, TURNOVER® Lifecycle Manager looks for objects checked out under the task reference you specify, and then displays two lists of checkout references, one for non-IFS objects, and one for IFS objects. You can use these lists to select the items you want to add to the form.

- 3 Next, see *'Adding IFS objects to a form' on page 211*.

## Summary of IFS functions in Form Maintenance

| Function description                          | Implementation                                                                                 |
|-----------------------------------------------|------------------------------------------------------------------------------------------------|
| Toggle in and out of IFS mode                 | F8                                                                                             |
| Add an IFS form line (in IFS mode)            | F6<br><br>This uses the <code>TADDIFSLIN</code> command.                                       |
| Link commands to IFS form lines (in IFS mode) | Select an IFS line with Option 11 or 12.<br><br>This uses the <code>TADDIFSCMD</code> command. |
| List checked out IFS objects (in IFS mode)    | F19                                                                                            |

## Command line

Run the `ADDFORMI` command.

## Program

To use another application to create the form, you can write a program that issues the `ADDFORM` or `ADDFORMI` command.

If you use `ADDFORMI`, your program can run from a command line or from within another application program.

## Adding IFS objects to a form

### Using the interface

If you created a form by using F6 from Work with Forms, TURNOVER® Lifecycle Manager adds lines, based on the task reference. When TURNOVER® Lifecycle Manager displays the lists of checked out objects, select the ones you want on the form and press Enter.

TURNOVER® Lifecycle Manager builds the form, and then you can link commands to specific form lines, if required: see [‘Linking pre-run and post-run commands to IFS form lines’ on page 212](#).

The Work with IFS Checkout References panel appears during the form build procedure, and when you press F19 in IFS mode of Form Maintenance.

|                                                                                                                                   |                                                  |              |
|-----------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|--------------|
| <b>MM/DD/YY</b>                                                                                                                   | Work with IFS Checkout References                | Your Company |
| 16:49:42                                                                                                                          |                                                  |              |
|                                                                                                                                   | Form . . . 5103387 Update documentation for IFS. |              |
|                                                                                                                                   | Reference . . . PROJ0003                         |              |
| The objects listed below have been checked out using the above reference number but have not yet been included on the above form. |                                                  |              |
| 1=Include on form (To display full path, put cursor on object and press F22.)                                                     |                                                  |              |
| Type                                                                                                                              | Object                                           |              |
| <u>1</u> IFSDOC                                                                                                                   | /DOCS/TURNOVER/PUBLISH/R51/IFSMAN.DOC            |              |
| <u>1</u> IFSDOC                                                                                                                   | /DOCS/TURNOVER/PUBLISH/R51/TROUBLE.DOC           |              |
| <u>1</u> IFSDOC                                                                                                                   | /DOCS/TURNOVER/PUBLISH/R51/SETUP.DOC             |              |
| <u>1</u> IFSDOC                                                                                                                   | /DOCS/TURNOVER/PUBLISH/R51/INFO.DOC              |              |
| <u>1</u> IFSDOC                                                                                                                   | /DOCS/TURNOVER/PUBLISH/R51/FILES.DOC             |              |
| <u>1</u> IFSDOC                                                                                                                   | /DOCS/TURNOVER/PUBLISH/R51/EXITS.DOC             |              |
| <u>1</u> IFSDOC                                                                                                                   | /DOCS/TURNOVER/PUBLISH/R51/DISTRIB.DOC           |              |
| <u>1</u> IFSDOC                                                                                                                   | /DOCS/TURNOVER/PUBLISH/R51/AUTOMARK.DOC          |              |
| <u>1</u> IFSDOC                                                                                                                   | /DOCS/TURNOVER/PUBLISH/R51/OVERVIEW.DOC          |              |
| F3=Exit F11=Show this level F12=Cancel F21=Select all F22=Full path                                                               |                                                  |              |

If the form contains both IFS and non-IFS objects, then you see a checkout references panel for each set during the F6 form build procedure. The panel for regular objects appears first, followed by the IFS objects panel.

## Command line

Run the `TADDIFSLNI` command to add IFS lines one at a time.

## Program

To add multiple items at once, your program can issue the `TADDIFSLIN` or `TADDIFSLNI` command for each object you want to add.

If you use `TADDIFSLNI`, your program can run from a command line or from within another application program.

---

## Linking pre-run and post-run commands to IFS form lines

With IFS objects, just as with other kinds of objects, you can issue commands during a form run to further automate your object management activities or customize them in response to specific situations.

Commands linked to IFS form lines run either immediately before or immediately after the IFS object is processed.

You can link commands:

- using the interface, from Form Maintenance
- from a command line
- globally, by type code. See *'Optimizing Java programs' on page 223*.

Most often, you do this from Form Maintenance as you add IFS lines.



## Using the interface

| MM/DD/YY                                                              | Form Maintenance - IFS |                               | Your Company            |
|-----------------------------------------------------------------------|------------------------|-------------------------------|-------------------------|
| 16:30:13                                                              |                        |                               |                         |
| Form . . . . .                                                        | 5103387                | Update documentation for IFS. |                         |
| Application . . . .                                                   | DOC                    | Rel/Ver/Lev . . . . .         | 5/01/01                 |
| Job description . .                                                   | QBATCH                 | Output queue . . . . .        | MARYSOQ                 |
| Library . . . . .                                                     | QGPL                   | Library . . . . .             | MARYS                   |
| Library list . . . .                                                  | DOC050101              | Job accounting . . . .        |                         |
| Programmer . . . . .                                                  | MARYS                  | Reference . . . . .           | PROJ0003                |
| Form comments: N Lock checks: N Completion messages: Y                |                        |                               |                         |
| Position to line . . . ____ or Object . . . ____                      |                        |                               |                         |
| 2=Change 4=Delete M=Move B=Before A=After 11=PreRunCmd 12=PostRunCmd  |                        |                               |                         |
|                                                                       |                        |                               |                         |
| Ln                                                                    | Object                 | Type                          | Cd Path                 |
| <u>11</u>                                                             | 001 HOWTO.DOC          | IFSDOC                        | A /DOCS/TURNOVER/REVIEW |
|                                                                       |                        |                               |                         |
| <b>Bottom</b>                                                         |                        |                               |                         |
| F3=Exit F4=Comments F5=Resequene F6=Add line F12=Cancel F24=More keys |                        |                               |                         |

| MM/DD/YY                                                 | Add IFS-Linked Command |  | Your Company |
|----------------------------------------------------------|------------------------|--|--------------|
| 16:38:31                                                 |                        |  |              |
| Form . . . . .                                           | 5103387                |  |              |
| Command type . . .                                       | Pre-Run                |  |              |
| Sequence . . . . .                                       | 1                      |  |              |
| Type command, press Enter.                               |                        |  |              |
| Command to execute: <u>type your command here...</u>     |                        |  |              |
| _____                                                    |                        |  |              |
| _____                                                    |                        |  |              |
| _____                                                    |                        |  |              |
| Error recovery: C C=Cancel (end job) G=Go (continue job) |                        |  |              |
| Link to line: <u>1</u> Object: HOWTO.DOC                 |                        |  |              |
| F10=Select Path: /DOCS/TURNOVER/REVIEW                   |                        |  |              |
| Type: IFSDOC                                             |                        |  |              |
| Library list . . . . <u>*CURRENT</u> *APPL, *CURRENT     |                        |  |              |
| F3=Exit F4=Prompt F9=Retrieve F10=Select F12=Cancel      |                        |  |              |

## Command line

Run the TADDIFSCMI command. Press F10 to see all the parameters.

## About writing programs for IFS-related form processing

If you are dealing with a large number of objects, and want to write a program that issues the ADDFORM, TADDIFSLIN or TADDIFSCMD command to add lines and link commands, there are some things you should keep in mind.

- These commands use an additional parameter to return information to the calling program.
- You do not have to use the parameter, but your program must define it.

- If you are not using the additional parameter, make sure your program monitors for and ignores the resulting error message.

---

## Running IFS forms

- 1 On the Work with Forms panel, select your form with option 7.

TURNOVER® Lifecycle Manager processes the objects as your application definition and type codes indicate.

- 2 When the form job completes, check the form logs, and make sure that your IFS objects are where you expected them to be.

---

## Distributing IFS objects

Distributing IFS objects is much like distributing other IBM i objects, except it uses directories instead of libraries.

### Distribution-related subdirectories for IFS objects

The first time that you run a promotion that includes distributing IFS objects, TURNOVER® Lifecycle Manager looks for an IFS directory called `TURNOVER`. If the directory does not already exist, TURNOVER® Lifecycle Manager creates it.

In the `TURNOVER` directory, TURNOVER® Lifecycle Manager creates these subdirectories for current and future use during IFS distribution processing:

`DISTRIBUTION`

This is the distribution work directory. When a distribution job runs that includes IFS objects, the following subdirectories are created under it:

`DISnnnnnnn`

Named for the distribution job associated with the form.

`ffffff`

Named for the form number containing IFS objects for distribution.

Distributed objects are created in the `ffffff` subdirectory as `lll`, where `lll` is the object's form line number.

For example, the IFS object described by IFS line 4 of form 5103389, which ran under distribution job DIS007252, appears in the IFS distribution directory as:

`/TURNOVER/DISTRIBUTION/DIS0007252/5103389/004`

## RECEIVE

This is the receive form directory. This directory is created on remote computers where a production computer has sent one or more forms for installing. When the remote computer receives a form that contains IFS objects, a subdirectory and objects are created here as follows:

*ffffff*

Named for the form number containing the IFS objects that were distributed to the remote computer.

Received objects are created and named *lll*, where *lll* = the IFS line number.

For example, the IFS object described by IFS line 4 of form 5103389 received from the production computer appears in the IFS receive directory as:

/TURNOVER/RECEIVE/5103389/004

## Other preparations for distributing IFS objects

- 1 Make sure the appropriate IFS application definitions have been created on your remote computers.
- 2 Make sure your application's IFS type codes are set up to distribute objects, as needed

---

## Finding TURNOVER® Lifecycle Manager information about IFS objects

TURNOVER® Lifecycle Manager records the activities it performs on any object. You can track the promotion, archive, distribution, or checkout history for an IFS object.

---

### IFS object history

To see a list of IFS objects TURNOVER® Lifecycle Manager has promoted, use main menu option 4, and then press F8.

The Work with IFS Object History panel appears.

|                                                                                                                                             |                                     |                    |                    |
|---------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|--------------------|--------------------|
| MM/DD/YY<br>10:54:46                                                                                                                        | <b>Work with IFS Object History</b> |                    | T51QA<br>SLS       |
| To reposition screen, enter Object:                                                                                                         |                                     |                    |                    |
| Filter by Type:                                                                                                                             | Programmer:                         | Appl:              |                    |
| Status:                                                                                                                                     | Reference:                          |                    |                    |
| 3=Copy archived object 5=View form 6=Explode paths 8=Archive information<br>9=View task 10=Checkout history 11=Line details 16=Dist History |                                     |                    |                    |
| Object                                                                                                                                      | Type                                | Path               | Cd Status Run Date |
| ___ AUTOMARK.DOC                                                                                                                            | IFSDOC                              | ...RNOVER/REVIEW A | Ready 0/00/00      |
| Reference: PROJ0003 Pmgr: MARYS Archived: N Checked out: N                                                                                  |                                     |                    |                    |
| Form: 5103389 Appl: DOC Rel: 5 Ver: 1 Lev: 1                                                                                                |                                     |                    |                    |
| ___ DISTRIB.DOC                                                                                                                             | IFSDOC                              | ...RNOVER/REVIEW A | Ready 0/00/00      |
| Reference: PROJ0003 Pmgr: MARYS Archived: N Checked out: N                                                                                  |                                     |                    |                    |
| Form: 5103389 Appl: DOC Rel: 5 Ver: 1 Lev: 1                                                                                                |                                     |                    |                    |
| ___ EXITS.DOC                                                                                                                               | IFSDOC                              | ...RNOVER/REVIEW A | Ready 0/00/00      |
| Reference: PROJ0003 Pmgr: MARYS Archived: N Checked out: N                                                                                  |                                     |                    |                    |
| Form: 5103389 Appl: DOC Rel: 5 Ver: 1 Lev: 1                                                                                                |                                     |                    |                    |
| More...                                                                                                                                     |                                     |                    |                    |
| F3=Exit F4=List                                                                                                                             | F11=Show all                        | F12=Cancel         |                    |
| F21=System command F22=Full path                                                                                                            |                                     |                    |                    |

This panel displays the most recent information for each object. It includes only objects only for the which applications you are authorized to use.

| To                                                              | Do this                                                                                     |
|-----------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| find out when and where an object was archived                  | Select it with option 8, and then see <a href="#">'Archive information' on page 216</a> .   |
| display the checkout history                                    | Select it with option 10, and then see <a href="#">'Checkout history' on page 217</a> .     |
| display the systems to which an IFS object has been distributed | Select it with option 16, and then see <a href="#">'Distribution history' on page 217</a> . |

### Archive subdirectories for IFS objects

TURNOVER® Lifecycle Manager stores archived objects in the /TURNOVER/ARCHIVE subdirectory.

In addition, you can specify in your application definition a default archive subdirectory that your application should use when archiving objects. This value is set using the TURNOVER® Lifecycle Manager global default **IFS Archive subdirectory**. The default value is DEFAULT. You can override the global default for any level of an application by specifying a different value on the Application defaults and rules panel for that application level.

### Archive information

To display archive information about an object, use option 8.

| MM/DD/YY                                  |         | Work with Archived IFS Objects |                 |        |              | T51QA          |  |
|-------------------------------------------|---------|--------------------------------|-----------------|--------|--------------|----------------|--|
| Object . . . <b>DISTRIB.DOC</b>           |         |                                |                 |        |              |                |  |
| Path . . . . <b>/DOCS/TURNOVER/REVIEW</b> |         |                                |                 |        |              |                |  |
| 1=Restore 3=Copy                          |         |                                |                 |        |              |                |  |
|                                           | Form    | Line                           | Archive<br>Date | Online | Save<br>Date | Save<br>Volume |  |
| _                                         | 5103391 | 001                            | 12/12/19        | Y      |              |                |  |
| _                                         | 5103391 | 002                            | 12/12/19        | Y      |              |                |  |
| _                                         | 5103214 | 001                            | 03/08/19        | N      | 04/15/19     | SLS210         |  |
| _                                         | 5103214 | 002                            | 03/08/19        | N      | 04/15/19     | SLS210         |  |
| _                                         | 5103220 | 002                            | 05/19/19        | Y      | 07/15/19     | SLS211         |  |
| F3=Exit F12=Cancel                        |         |                                |                 |        |              |                |  |

An object can be saved offline and still exist online. Any object that exists online is eligible for the Copy option, whether it has been saved offline or not. An object that has been saved offline and no longer exists online is eligible only for the Restore option.

## Checkout history

To display the checkout history for an IFS object, use option 10.

| MM/DD/YY                                  |  | Display Checkout History |  |                                  |  | T51QA                       |  |
|-------------------------------------------|--|--------------------------|--|----------------------------------|--|-----------------------------|--|
| Object . . . <b>OVERVIEW.DOC</b>          |  |                          |  |                                  |  |                             |  |
| Path . . . . <b>/DOCS/TURNOVER/REVIEW</b> |  |                          |  |                                  |  |                             |  |
| Position to Date . . . . _____            |  |                          |  | Filter by Programmer . . . _____ |  |                             |  |
|                                           |  |                          |  | Reference . . . _____            |  |                             |  |
| <b>Development Path</b>                   |  | <b>Check-out</b>         |  | <b>Check-in</b>                  |  | <b>Programmer Reference</b> |  |
| /DOCS/TURNOVER/WORK                       |  | 12/03/98 14:18           |  | 12/04/98 8:29                    |  | MARYS PROJ0003              |  |
| F3=Exit F12=Cancel F22=Full path          |  |                          |  |                                  |  |                             |  |

## Distribution history

To display the systems to which an IFS object has been distributed, use option 16.

```

MM/DD/YY                Maintain Form Distribution Status                T51QA
12:42:59

Form . . . . . 5103375
Description . . . IFS Documentation Development
Status . . . . . RAN-OK

2=Change   8=Work w/ Distribution Job

Distribution
Sys System  Description      Job          Date    Dist Device  Confirmed
150 WASHDC   SLS Washington DC   DIS5100074  05/19/19  Y   *SNADS    Y
_ 410 HOUSTON SLS Houston        DIS5100074  05/19/19  Y   *SNADS    Y
_ 420 SEATTLE SLS Seattle        DIS5100074  05/19/19  Y   *SNADS    Y
_ 430 ATLANTA SLS Atlanta        DIS5100074  05/19/19  Y   *SNADS    Y

F3=Exit   F12=Cancel
  
```

## Checked out IFS objects

### To display a list of the IFS objects that are checked out

On the main menu, select option 5. On the Checkout/Checkin Source menu, select option 36.

The View Checked Out IFS Objects panel appears.

```

MM/DD/YY                View Checked Out IFS Objects                T51QA

Position to object: _____ Filter by Programmer: MARYS _____
                                Reference: _____
                                Appl/Rel/Ver/Lev: _____

2=Change checkout record

Object                      Type      Programmer Reference
AUTOMARK.DOC                IFSDOC   MARYS      PROJ0003
  Path: /DOCS/TURNOVER/PUBLISH/R51
  Application: DOC Release: 05 Version: 01 Level: 02 Exclusive: Y
_ DISTRIB.DOC                IFSDOC   MARYS      PROJ0003
  Path: /DOCS/TURNOVER/PUBLISH/R51
  Application: DOC Release: 05 Version: 01 Level: 02 Exclusive: Y
_ EXITS.DOC                  IFSDOC   MARYS      PROJ0003
  Path: /DOCS/TURNOVER/PUBLISH/R51
  Application: DOC Release: 05 Version: 01 Level: 02 Exclusive: Y
_ 2 FILES.DOC                IFSDOC   MARYS      PROJ0003
  Path: /DOCS/TURNOVER/PUBLISH/R51
  Application: DOC Release: 05 Version: 01 Level: 02 Exclusive: +

F3=Exit   F4=Reference selection   F12=Cancel   F22=Full path/object
  
```

Option 2 (Change checkout record) enables you to change the type code if you have checked out the object under the wrong type code.

| MM/DD/YY                     |  | Change Checkout Record                |  | T51QA                         |  |
|------------------------------|--|---------------------------------------|--|-------------------------------|--|
| 12:58:39                     |  | Application: DOC Rel: 5 Ver: 1 Lev: 2 |  |                               |  |
| Object . . . . .             |  | FILES.DOC                             |  |                               |  |
| Type code . . . . .          |  | IFSDOC                                |  |                               |  |
| Checked out from path . . .  |  | /DOCS/TURNOVER/PUBLISH/R51            |  |                               |  |
| Development path . . . . .   |  | /DOCS/TURNOVER/WORK                   |  |                               |  |
| Programmer . . . . .         |  | MARYS                                 |  |                               |  |
| Reference . . . . .          |  | PROJ0003                              |  |                               |  |
| New type code . . . . .      |  | *SAME                                 |  | *SAME, Type code, F4=List     |  |
| New programmer . . . . .     |  | *SAME                                 |  | *SAME, *USRPRF, Name, F4=List |  |
| New reference . . . . .      |  | *SAME                                 |  | *SAME, Char, F4=List          |  |
|                              |  | *SAME                                 |  |                               |  |
| Copy test object . . . . .   |  | N                                     |  | Y,N                           |  |
| Delete test object . . . . . |  | N                                     |  | Y,N                           |  |
| F3=Exit F4=List F12=Cancel   |  |                                       |  |                               |  |

## IFS reference

### IFS file summary

TURNOVER® Lifecycle Manager records information about IFS objects in the following files:

| File name | Description                    |
|-----------|--------------------------------|
| TIFSEXTF  | IFS Path Extension file        |
| TIFSDFTF  | IFS Application Defaults file  |
| TIFSAUTF  | IFS Authority List Object file |
| TIFSTYPF  | IFS Application Type Code file |
| TIFSEXPf  | IFS Application Explode List   |
| TIFSOUTF  | IFS Checkout file              |
| TIFSLINF  | IFS Form Line file             |
| TIFSCMDF  | IFS Linked Commands file       |
| TIFSARCF  | IFS Archive Index file         |
| TIFSOFFF  | IFS Offline Archive Index file |
| TIFSCONF  | IFS Conflict file              |

---

## IFS exit points

Several exit points that are provided so you can customize TURNOVER® Lifecycle Manager' IFS processing.

### To view the exits for IFS

- 1 On the main menu, select option 8, and then select option 7.
- 2 On the Work with Exits panel, press F8.

| To                                            | Do this                                                                |
|-----------------------------------------------|------------------------------------------------------------------------|
| display more information about an exist       | Select it with option 8.                                               |
| display an example program source code member | Select it with option 7, and then choose the associated source member. |

- Copy the sample source before modifying it to avoid loss of the original.
- Cross-reference information is not currently supported for IFS objects. However, you might use one or more of these exit points to perform some of your own cross-referencing.

### IFS 2: Post checkout exit

Exit IFS 2 is called after an IFS object has been checked out and copied to your development path.

#### Example

This exit is useful when working with IFS objects that are associated with supplemental objects such as control files or help files. The exit program can check the IFS object name and, if it has dependent files, copy them to the development path. You might also use your program to raise a flag or send a message when company-sensitive objects are checked out.

### IFS 3: Manual check-in exit

Exit IFS 3 is called when you manually check in an IFS object. Use this exit to perform additional check-in security processing, and/or other custom processing.

With manual check-in, the specified IFS object does not replace the production object. Manual check-in frees the lock on the object so that someone else can check it out.

When a user tries to check in an IFS object manually, TURNOVER® Lifecycle Manager makes sure that the user has check-in authority in general and for the particular application. If the user does not have check-in authority, TURNOVER® Lifecycle Manager does not allow the object to be checked in.



### Example 1

You might want to perform a more extensive verification procedure. For example, you might allow the programmer who checked out the object to check it in. Furthermore, you might want to allow that programmer to check in only objects that have a particular attribute.

### Example 2

You can use IFS Exit 3 to perform other tasks when an object is checked in. For example, perhaps a particular object is associated with control or help files that are generally checked out with the object. Your IFS Exit 3 program could search the path for these files, and if they exist, automatically check them in as well.

## IFS 4: Add line exit

Exit IFS 4 is called whenever an IFS line is added to a form, whether implicitly or explicitly. You might decide to change or evaluate one or more of the many parameters passed in to the exit program.

Any of the parameter values changed within the exit program take precedence over those defined during the application default process.

### Example

For example, when an IFS line is added to a form, you might want to check the associated action code. If it happens to be a D, indicating object deletion, you can send a message to the users on the form approval list, as well as to QSECOFR, warning them of the pending deletion.

## IFS 5: Pre-checkout exit

Exit IFS 5 is called before IFS objects are checked out and copied to your development path. Given the list of parameters that are passed to the exit program, there are many different conditions you can evaluate to determine whether or not to allow a checkout request to proceed.

The last parameter passed, `&CHECKOUT`, is a flag that can be modified by your IFS Exit 5 program. If one of your conditions for checkout fails, you can set this flag to `N` to cancel the checkout request.

For parameter `&CHECKOUT`: Y = success; N = failure.

As part of the standard TURNOVER® Lifecycle Manager checkout process, these conditions are already evaluated:

- TURNOVER® Lifecycle Manager verifies that the user requesting a given member/object has general checkout authority, as well as authority for the particular application.
- TURNOVER® Lifecycle Manager ensures that the requested object is not already checked out.

You might want to go beyond the standard TURNOVER® Lifecycle Manager evaluation process for checkout.

### Example

For example, you might want to allow only users belonging to a specific group profile to check out an object.

### IFS 7: Post add line exit

Exit IFS 7 is called after a line is added to a form.

If your exit program must change the parameters to the line that is being added, use [IFS 4: Add line exit' on page 221](#) instead of exit IFS 7.

However, if your processing requires a TURNOVER® Lifecycle Manager line number, exit IFS 7, because the line number is not yet available when exit IFS 4 is called.

### Example

You might use this exit if you need to update a database to reflect the form and line number associated with the object being added to the form.

---

## Troubleshooting IFS support

**I checked an existing IFS object out of the production folder. The confirmation message said "Object checked out," but when I look in my development folder, the object isn't there. Why not?**

TURNOVER® Lifecycle Manager could not find the IFS object in the production folder, so it reserved a new object in the specified path and checked it out.

If TURNOVER® Lifecycle Manager finds an IFS object in the production folder, it copies it, and you see "Object copied." If the object is not found in production, it is considered new, and you see "Object checked out."

To solve the problem, determine why TURNOVER® Lifecycle Manager could not see your object. Usually, this is caused by a discrepancy in the path you expected or supplied, as opposed to the one specified in the type code for that application level. The application type code remembers the appropriate IFS paths for each level of your application; all you need to supply on the checkout panel is the object name. If you specified the full path for your object on the checkout panel, TURNOVER® Lifecycle Manager looked for those directories within the path it already knew about, and that is why your object was not found.

**My IFS processing failed because of insufficient authority. (This can happen during checkout or form processing.)**

The **IFS Adopted authority user** global default identifies the user profile from which TURNOVER® Lifecycle Manager should adopt authority for managing IFS objects. Its default value is \*CURRENT, which means that the user profile of the user doing the work is used. If that user profile does not have \*ALLOBJ authority, then TURNOVER® Lifecycle Manager jobs that manage IFS objects to which the user does not have authority will fail.

Create a user profile that has \*ALLOBJ authority on your system, and specify that profile for the **IFS Adopted authority user** global default.

To manage objects with the TURNOVER® Lifecycle Manager Client (on Windows), you must create an identical Windows user and password, and give this user full permission to any network shares you plan to manage using the TURNOVER® Lifecycle Manager Client.

For more information about managing IFS objects using the TURNOVER® Lifecycle Manager Client, see *'To manage these values for an IBM i user profile, use the User ID number and Group ID number parameters of the CRTUSPRF, CHGUSRPRF or DSPUSRPRF IBM i commands. (To see these fields, page down about 5 times on the command prompt.)' on page 232.*

**There are no objects in the /TURNOVER/RECEIVE directory on my remote computer. How can I find out if TURNOVER® Lifecycle Manager successfully packaged the IFS objects on my form before distributing?**

TURNOVER® Lifecycle Manager packages IFS objects in their own individual save file, named `IFSnnnnnn`, where `nnnnnn` is the job number associated with the distribution job. This save file is in turn packaged in the save file that contains the form and any non-IFS objects associated with it.

Check the distribution joblog on the production system for error messages pertaining to the creation of an `IFSnnnnnn` save file.

---

## Optimizing Java programs

You can expand the basic IFS support with a special automatic linked command feature that lets you build Java programs from CLASS, JAR, or ZIP files during promotion.

### Overview

This feature enables you to create and sustain a default linkage between CLASS, ZIP, or JAR object types on your TURNOVER® Lifecycle Manager form to the IBM i `CRTJVAPGM` command. The `CRTJVAPGM` command builds a hidden IBM i Java program that becomes part of the CLASS, ZIP, or JAR file and runs on the IBM i when the class is started by the IBM i `JAVA` command.

You do not have to create the hidden commands when you promote these kinds of files. If they do not exist, the IBM i `JAVA` command build them (depending on the value of its Optimization parameter). However, creating has these benefits:

- improved initial runtime for the class
- greater control over special creation parameters (such as `USRPRF` and `USEADPAUT`, which are not available on the `JAVA` command; to specify values for them you can override the command in the application type code definition, or edit the form line.)
- greater control over optimization parameters.

## Configuring default linked commands

Familiarize yourself with the `CRTJVAPGM` command and its parameters. Understand the values that you want to specify and study the values that are already linked to the `CLASS` global type code.

To configure TURNOVER® Lifecycle Manager for specialized processing of Java files, follow these instructions.

### 1: Check your global CLASS (or JAR or ZIP) type code

Edit the global type code with option 2.

- If you are a new TURNOVER® Lifecycle Manager user, an existing `CLASS` type code might already have the `CRTJVAPGM` command linked as a default post-line command. Verify the command parameters, and then change them if necessary.
- If you upgraded from a previous release of TURNOVER® Lifecycle Manager, your type codes were preserved during the upgrade. You can add the default linked command to an existing IFS type code. In edit mode, press F6.

|                                                                          |                                   |                    |
|--------------------------------------------------------------------------|-----------------------------------|--------------------|
| MM/DD/YY                                                                 | Add IFS Type Code Default Command | Your Company, Inc. |
| Type code . . . . CLASS                                                  |                                   |                    |
| Command type . . . * POSTRUN *PRERUN, *POSTRUN                           |                                   |                    |
| Sequence . . . . 10                                                      |                                   |                    |
| Type command, press ENTER                                                |                                   |                    |
| Command to execute: CRTJVAPGM CLSF('&PO') OPTIMIZE(30) REPLACE(*YES)_    |                                   |                    |
| <hr/>                                                                    |                                   |                    |
| &TY=Type code                                                            | &FP=From path                     | &PG=Programmer     |
| &TP=Target path                                                          | &FO=From object                   | &RF=Reference      |
| &OB=Target object                                                        | &ME=Method                        | &AL=Authority list |
| <hr/>                                                                    |                                   |                    |
| Error recovery: C Cancel (end Turnover job) G=GO (continue Turnover job) |                                   |                    |
| Library list . . . . *CURRENT *@APPL, *CURRENT                           |                                   |                    |
| Run on system . . . . *ALL *ALL, *LOCAL, *REMOTE, Name                   |                                   |                    |
| F3=Exit F4=Prompt F12=Cancel                                             |                                   |                    |

If no global IFS type code exists for the purpose, press F6 on the Work with Type Codes panel to add a global type code, and then press F8 to add an IFS type code. Fill in the necessary information, and then exit. Edit the type code, and then add the command by using F6 as described above.

## Turning on automatic linking for applications

- 1 Include the global type code in your application at all levels.
- 2 To turn on the automatic include feature, set **Add default linked commands** to Y.

```

MM/DD/YY                CLASS Defaults                Your Company, Inc.
08:33:04      Application: WEB  Rel:  1 Ver:  1 Lev:  1      SYSTEM: YOURSYS

Type choices and press Enter.

Target path . . . /website/qa                                F22=Full path      Locked: N

From path . . . . /website/dev                                F22=Full path      Locked: N

Add default linked commands . . Y Y, N
Promotion method . . . . . CPY CPY, MOV, *NULL
  Locked . . . . . N Y, N
Authority list . . . . . WEBPROD F4=List
  Locked . . . . . Y Y, N
Delete from object . . . . . N Y, N
  Locked . . . . . N Y, N
Archive . . . . . Y Y, N
  Locked . . . . . Y Y, N
Explode . . . . . N Y, N      Deploy . . . N Y, N
Distribute . . . . . Y Y, N
Development authority list . . . WEBDEV F4=List
F3=Exit F4=Select authority list F12=Cancel F22=Full path

```

## Checking the setup

- 1 Copy an existing CLASS file that you can use as a test.

Store the file in the production directory identified by the CLASS type code at your application's highest level.

- 2 Choose a project and task reference which you can use to check out the test class file.
- 3 Check out the CLASS file. Do one of the following:
  - On the Check-in/Checkout menu, select option 11.
  - On the command line, use the TCHKOUTIFS command.
- 4 Add the CLASS file to a form using F6 from Work with Forms.

The Form Maintenance panel appears.

```

MM/DD/YY                Form Maintenance - IFS                Your Company
11:22:28

Form . . . . . 5300484      Fix broken class file
Application . . . . WEB      Rel/Ver/Lev . . . . . 1/01/01
Job description . . QPGMR      Output queue . . . . . PGMR10Q
  Library . . . . . QGPL      Library . . . . . PGMR1
Library list . . . . WEB010101 Job accounting . . . .
Programmer . . . . . PGMR1      Reference . . . . . WEB 0003
Form comments: N Lock checks: Y Completion messages: N
Position to line . . . ____ or Object . . . _____
2=Change 4=Delete          M=Move B=Before A=After 11=PreRunCmd 12=PostRunCmd

Ln Object                                Type      Cd Path
__ 001 myclass.class                    CLASS     A  /web/qa
2 CRTJVAJPGM CLSF('&PO') OPTIMIZE(30) REPLACE(*YES)

Bottom
F3=Exit F4=Comments F5=Resequenece F6=Add line F12=Cancel F24=More keys

```

- 5 Use option 2 to edit the CRTJVAPGM command, and then change its parameter values if necessary.
- 6 Run the form.
- 7 Monitor the job using option 8 in Form Maintenance.

Messages issued by the CRTJVAPGM command are written to the log report for the form job.

---

## Promoting files in the Windows file system

TURNOVER® Lifecycle Manager can manage PC objects directly by using the “Windows file system” in IFS support on IBM i. The Windows file system enables a program on an IBM i computer to see workstation computers that are connected to the same network.

### System requirements

- IBM i V7.1
- Windows 2000, or higher, on the client PC.

### Prerequisites

To access PCs in your Windows network, configure and start IBM i NetServer on the IBM i computer. For more information about NetServer, see:

<http://www-03.ibm.com/systems/power/software/i/netserver/>

---

## Configuring for Windows file management

- 1 On the Windows network, create a user with the same name and password as the user specified as the Adopted IFS authority user in TURNOVER® Lifecycle Manager global defaults).
- 2 Share the appropriate Windows directories on the client computer, and give full control permissions to the IFS Adopted authority user profile on the Windows network.

From Windows, you would access these shared folders from another computer by navigating there through direct file specification, something like this:

```
\\SystemName\ShareName\SubDir\Filename.ext
```

- 3 Configure your TURNOVER® Lifecycle Manager application definition type codes with the correct path statements.

From the IBM i computer, you can access the same shared folders by using a path statement similar to this:

```
/QNTC/SystemName/ShareName/SubDir/Filename.ext
```

---

## Troubleshooting Windows support

- Limit the IFS adopted authority user's access to only those folders you want TURNOVER® Lifecycle Manager to manage.
- Run a test using your reconfigured TURNOVER® Lifecycle Manager application before you begin managing live files.

### To access workstation computers in the Windows Network

Do one of the following:

- Use the IBM i `WRKLNK` command:

```
WRKLNK OBJ ('/QNTC/*')
```

- On DATA menu, select option 5 (Integrated File System commands), then option 2 (Object commands), and then option 1 (Work with object links). For the object parameter, specify `/QNTC/*`, and then press Enter.

This command produces a list of all the workstation computers the IBM i can see on the network.

If you have shared the appropriate directories on the computer you want TURNOVER® Lifecycle Manager to access, but it does not appear in the IBM i computer's list of accessible workstations, you can make it available by running the IBM i `MKDIR` command. For example:

```
MKDIR DIR ('/QNTC/WindowsSystemName')
```

*WindowsSystemName* is the name of the workstation computer as it appears in the Windows Network Neighborhood.

---

## Promoting files in a UNIX file system

TURNOVER® Lifecycle Manager can manage files that reside in the Integrated File System (IFS). The IBM i IFS enables the IBM i system to recognize objects from other operating system file systems. These file systems can be on the IBM i computer itself or on other computers on the network. It does not matter to the IBM i system where the files are, as long as it can access them.

TURNOVER® Lifecycle Manager can access and manipulate any file that the IBM i can manage. The features and limitations of TURNOVER® Lifecycle Manager mirror those of the IBM i APIs that it uses.

IBM i supports the Network File System (NFS) which is also supported by UNIX.

NFS is a client/server protocol: the IBM i computer which TURNOVER® Lifecycle Manager is on is the client, and the UNIX computer is the server.

### Operating systems

This feature works with any operating system that provides a Network File System (NFS) server, including any version of UNIX. It has been tested with the following operating systems:

- RedHat Linux 7.1
- Mac OS X 10.1.2

## Additional information

<http://www.redbooks.ibm.com/abstracts/sg246551.html?Open>

---

## Configuring TURNOVER® Lifecycle Manager support for UNIX

### 1: Define exports on the UNIX systems

On the UNIX system you must define the directories that you want to make available to the IBM i system. In UNIX terminology, this is called an “export.”

- 1 To share a UNIX directory with other systems, you must export it.

For example, On Linux systems, you do this by editing a file named `/etc/exports`. This file that contains a list of all of the directories you want to share, as well as some options that control how they are shared, and to whom. For example:

```
/opt/WebSphere/AppServer/installedApps as400.mycompany.com(rw,no_root_squash)
/pub *(ro,all_squash) # Public dir
/home *.mycompany.com(rw,squash uids=0-99) # User's Home directories
```

In this example, the file contains entries for three directories. Each line comprises three parts.

- Part 1 is the directory you are sharing; in this example:

`/opt/WebSphere/AppServer/installedApps.`

Because this Linux system is running WebSphere, we want to share the directory where our WebSphere applications are installed, so that we can use TURNOVER® Lifecycle Manager to manage them.

- Part 2 specifies the systems to which this export should be accessible.

To specify a system, use either a TCP/IP name (as the example shows), one IP address, or a range of IP addresses. For example, `192.168.0.0/255.255.255.0` allows any system in the IP address range of 192.168.0.1 - 192.168.0.254 to access the share.

- Part 3 defines some of the sharing options.

`rw` stands for read-write, and is necessary if TURNOVER® Lifecycle Manager is going to be able to install and change files.

`no_root_squash` indicates that the root user (that is, QSECOFR on an IBM i computer) is allowed to connect. TURNOVER® Lifecycle Manager needs to be able to connect as the root user to apply owner and permission settings, and to have the authority that is necessary to update the files. As long your QSECOFR profile is safe on the IBM i computer, this is not a major security concern.

- 2 Refresh the export definitions. On Linux, run this command:



```
/user/sbin/exportfs -a -v
```

## 2: Verify NFS service and firewall rules

Make sure that the NFS service is running on the UNIX system, and that network firewall rules do not deny access to the service.

- 1 To check the NFS service from the server, type:

```
ps ax | nfsd
```

A list of nfsd server processes should appear. If not, try starting the nfsd daemon by typing:

```
/etc/init.d/nfs start
```

The administrator can set up NFS to start permanently by using the SYSV Runlevel manager (`tksysv` command).

- 2 To view firewall rules, type:

```
ipchains -L
```

Only an authorized system administrator should change firewall rules.

## 3: Mount directories on the IBM i computer

Mount the exported file system on the IBM i computer to make the files and directories of the UNIX system available to the IBM i computer. You must mount the file system on top of an existing directory, which becomes the “mount point.”

All of the contents of that directory disappear and are replaced by the contents from the mounted file system. (When the file system is un-mounted, the contents reappear.)

- 1 Create a directory in the IFS root folder named `/NFS`, to contain all of the mount points. Then create one subdirectory for each UNIX system, and then create further subdirectories for each directory. For example:

```
/NFS
  /Linuxserver
    /WebSphere
    /Apache
  /AIXserver
    /Scripts
```

This example mounts three directories from two UNIX systems.

- 2 To mount a directory, use the IBM i `MOUNT` command. For example:

```

~                               Add Mounted FS (MOUNT)

Type choices, press Enter.

Type of file system . . . . . > *NFS          *NFS, *UDFS, *NETWARE
File system to mount . . . . . > 'linuxsys:/opt/WebSphere/AppServer/installed
Apps'
Directory to mount over . . . . > '/NFS/linuxsys/WebSphere'

Mount options . . . . . 'rw,suid,retry=5,rsz=8096,wsz=8096,timeo
=20,retrans=5,acregmin=30,acregmax=60,acdirmin=30,acdirmax=60,hard'

Code page:
  Data file code page . . . . . *BINARY      1-32767, *ASCII, *JOBCCSID...
  Path name code page . . . . . *ASCII       1-32767, *ASCII, *JOBCCSID

Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys

```

## Type of file system

Specify \*NFS.

## File system to mount

Use the format *hostname:filesystem*.

*hostname* can be either an IP address or a host name

*filesystem* must match what is in the export file.

## Directory to mount over

The local mount point.

## Mount options

The values shown above are the defaults; you should not need to change them.

*rw* mounts the file system as read/write.

*suid* indicates that UIDs (file owners) can be set.

- 3 After you have mounted the file system, its contents are available in the IBM i IFS as if it was on the IBM i computer.
- 4 To access it, use the WRKLNK command. For example:

```
WRKLNK OBJ('/NFS/linuxserver/WebSphere/*')
```

This command displays the contents of the `/opt/WebSphere/AppServer/installedApps` folder on the Linux system.

## 4: Configure TURNOVER® Lifecycle Manager global defaults

Configure TURNOVER® Lifecycle Manager as you would for IFS support, with these exceptions:

- 1 You must set the QNTC Adopted authority user global default. On the main menu, select option 8, and then select option 5. Go to page 2 of the defaults.

```

Set Global Defaults (SETGLOBE)

Type choices, press Enter.

Allow group checkout . . . . . > N          *SAME, Y, N
Use Delete Test Object rule . . > Y          *SAME, Y, N
Company name . . . . . > 'YOUR COMPANY, INC.'
Dft pre/post run cmd lib list . > *CURRENT  *SAME, *CURRENT, *APPL
Specify method of distribution > *DISTRIB   *SAME, *FORM, *DISTRIB
IFS Adopted authority user . . . > TURNIFS   Name, *SAME, *CURRENT
QNTC Adopted authority user . . > TURNIFS   Name, *SAME, *CURRENT
IFS Archive subdirectory . . . > DEFAULT    Character value
Source archive file . . . . . > ARCHIVESRC  Name, *SAME
Library name . . . . . > TURNARC          Name
Automatic form lock checking . . > *DTAOBJ   *SAME, *NONE, *FILE, *DTAOBJ

Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys

```

- 2 Set **IFS Adopted authority user** to QSECOFR.

This enables TURNOVER® Lifecycle Manager to connect to the NFS file system as the root user (QSECOFR on the IBM i computer), so it can set file ownerships and permissions.

If your application also manages objects in the QNTC file system, set **QNTC Adopted authority user** to the user TURNOVER® Lifecycle Manager should use when setting Windows file permissions and ownership.

## Promoting from UNIX to Windows or Windows to UNIX

TURNOVER® Lifecycle Manager copies or moves items directly, and there can only be one active user. Therefore, if your application will promote from UNIX to Windows or from Windows to UNIX, you must create a user called QSECOFR in your Windows domain.

## 5: Create a TURNOVER® Lifecycle Manager authority list for use by NFS systems

Use a TURNOVER® Lifecycle Manager authority list to permit the explicit list types of \*STD and \*UNIX.

- \*STD is used for objects that are in the native IBM i root file system.
- \*UNIX enables you to define UNIX-style permissions for the objects.

The definition looks like this:

|                               |                                  |                           |         |
|-------------------------------|----------------------------------|---------------------------|---------|
| MM/DD/YY                      | Change IFS Authority List Object |                           | YOURSYS |
| 11:38:03                      |                                  |                           |         |
| Type choices and press Enter. |                                  |                           |         |
| Authority list object         | NFSTEST                          |                           |         |
| Description                   | NFS Test                         |                           |         |
| Type                          | *UNIX                            | *STD,                     | *UNIX   |
| Owner (UID)                   | 0                                | 0-4294967294              |         |
| Group (GID)                   | 0                                | 0-4294967294              |         |
| Permissions:                  |                                  |                           |         |
| User/Owner                    | *RW                              | *NONE, *RWX, *RX, *RW,... |         |
| Group                         | *RW                              | *NONE, *RWX, *RX, *RW,... |         |
| Other/Public/World            | *RW                              | *NONE, *RWX, *RX, *RW,... |         |
| F3=Exit F12=Cancel            |                                  |                           |         |

You can specify the owner (UID) and group (GID) that you want to set on the objects you promote, and the permissions for the User, Group and Other.

TURNOVER® Lifecycle Manager sets these values using the `chown` and `chmod` APIs.

Create a \*UNIX authority list for as many levels of your application as is needed. For example, you might have one list for development, one for QA, and one for production.

### NFS owner GID and UID values and IBM i profiles

TURNOVER® Lifecycle Manager forms that manage UNIX objects RECOVER when replacing objects (action code R), if there exists on the IBM i no user profile with UID and GID values that match the NFS object owner on the NFS system.

Create IBM i profiles for all NFS object owners, and set the UID and GID values for those profiles to match their NFS counterparts.

To manage these values for an IBM i user profile, use the User ID number and Group ID number parameters of the `CRTUSRPRF`, `CHGUSRPRF` or `DSPUSRPRF` IBM i commands. (To see these fields, page down about 5 times on the command prompt.)

# 15 Mirroring systems

There are many reasons to mirror an IBM i system, and many mirroring packages that do this for you. This section contains a list of TURNOVER® Lifecycle Manager libraries and files that should be mirrored, so that the mirror system will be a functional TURNOVER® Lifecycle Manager system when it is needed.

On your system, TURNOVER® Lifecycle Manager might have been installed into libraries that are not named according to TURNOVER® Lifecycle Manager defaults, or the product might have been consolidated into a single library instead of three libraries (product, data, and language libraries).

## Do not mirror

- Data areas named `xxCHECK`, `xxCODE`, `CSCHEKxxxx`, and `CSCODExxxx`, where the `xs` represent a varying set of alphanumeric characters.
- Save files in the live product library (SOFTTURN by default); this locks the save files that are used during the distribution process on your live system; this can cause intermittent distribution failures.
- TURNOVER® Lifecycle Manager work libraries, as follows.

| Description                                      | Name                     |
|--------------------------------------------------|--------------------------|
| Form archive libraries                           | <code>TnnnnnnnnDa</code> |
| Form work libraries                              | <code>TnnnnnnnnW</code>  |
| Explode archive libraries                        | <code>Tnnnnnnnncc</code> |
| Explode recovery libraries                       | <code>Znnnnnnnncc</code> |
| TURNOVER® Data Manager explode staging libraries | <code>Xnnnnnnnncc</code> |
| Distribution staging libraries                   | <code>TnnnnnnnnS</code>  |
| Distribution save libraries                      | <code>Thhmmssjjj</code>  |

`nnnnnnnn` is the form number; `cc` is an alphanumeric counter; `a` is the ASP number (which can be blank); `hhmmss` is a time stamp; `jjj` is the associated job number.

## Mirror the TURNOVER® Lifecycle Manager data library

The default TURNOVER® Lifecycle Manager data library is SOFTTURND, and the whole library, or whatever library you use for TURNOVER® Lifecycle Manager data, should be mirrored, *except* for these:

| File name | Attribute | Default library | Type | Description           |
|-----------|-----------|-----------------|------|-----------------------|
| TDIST     | *FILE     | SOFTTURND       | PF   | Distribution defaults |
| TDISTH    | *FILE     | SOFTTURND       | PF   | Distribution history  |

Because your disaster recovery system must have these files in order for distribution to work correctly, use the CRTDUPOBJ and CLRPFM commands to create empty duplicates of those two existing files there. (This applies only if you have loaded TURNOVER® Lifecycle Manager using mirroring, as opposed to installing a fresh copy.)

## Mirror any TURNOVER® Lifecycle Manager interface libraries

Any TURNOVER® Lifecycle Manager interface library, such as TURNOVER® for Lansa for LANSA users, should also be mirrored.

## Mirror TURNOVER® Lifecycle Manager product job description and environment data

In your TURNOVER® Lifecycle Manager product library (SOFTTURN by default), mirror job descriptions, as well as the following environment files:

| File name | Attribute | Default library | Type | Description           |
|-----------|-----------|-----------------|------|-----------------------|
| TENVF     | *FILE     | SOFTTURN        | PF   | Environment File      |
| TUSRENVF  | *FILE     | SOFTTURN        | PF   | User Environment File |

## Mirror product data areas

In your TURNOVER® Lifecycle Manager product library, mirror the data areas, *except* for those mentioned in *'Do not mirror' on page 233*.

## Mirror your TURNOVER® Lifecycle Manager archive libraries

If your TURNOVER® Lifecycle Manager application definitions are set up to archive source, make sure that the archive libraries are included in your list of libraries to be mirrored.

## Using a mirrored system

To use the mirror system, you must apply a temporary authorization code for TURNOVER® Lifecycle Manager to function.

- You must be registered on [support.softlanding.com](https://support.softlanding.com) before you can obtain a temporary code.

- Because support site registration can be completed only during normal business hours, plan ahead and register now.
- On the mirror system, make sure that *all* remote computers, as well as the development computer, are defined in the list of system definitions in TURNOVER® Lifecycle Manager main menu option 11, then option 1.

# 16 Setting up the mail daemon

TURNOVER® Lifecycle Manager can send email messages when a selected event occurs in the project system. It puts an entry on a data queue (\*DTAQ) object with control information about the message. Another program monitors this \*DTAQ and builds and sends the message. Historically, this program had to run on Windows because that is where the email APIs that were needed to send the message resided. This has always been a problem or at least an inconvenience for many customers that do not want to maintain or run a Windows server. In addition, there are other problems such as making sure the program is restarted after the Windows system is rebooted or the IBM i server is IPL'd or being backed up.

For a solution that runs natively on the IBM i and can be managed by standard IBM i work management features, use the TURNOVER® Lifecycle Manager Mail Daemon. It monitors that same \*DTAQ and builds and sends the same email messages. You can run this program on your IBM i server and manage it just like any other job.

The mail daemon has most of the features that are in the existing Enhanced Mail Messenger. However there are some differences and considerations that you need to take into account. It also has extra features.

## Requirements

- TURNOVER® Lifecycle Manager 2.1 or later.
- A mail server that supports the SMTP (which just about all mail servers do). If your server has been set up to not allow relaying, change it allow relaying from at least the IP address of your IBM i server.
- The following IBM i Licensed Programs are required:

|         |       |                                     |
|---------|-------|-------------------------------------|
| 5770JV1 | *BASE | IBM Developer Kit for Java for Java |
| 5770JV1 | 16    | Java SE 8 32 bit                    |
| 5770JV1 | 17    | Java SE 8 64 bit                    |

- The system value QMLTTHDACN must be set to a value of 1 or 2. (A value of 2 is the shipped default from IBM.)



---

## Considerations for using the mail daemon

This section describes some differences between the Mail Daemon and the Windows Messenger. These differences involve documents, shortcuts, and URLs.

The TURNOVER® Client allows you to attach “links” to tasks. These links can be to any of the following:

- document attachments
- shortcuts to documents
- areas internal to TURNOVER® Lifecycle Manager such as tasks or requesters.

You can configure your email message so that these links are attached to the message.

### Shortcuts and URLs

When a shortcut or URL is to be attached to the message, the Windows messenger uses a special Windows API for creating a shortcut to a file or a URL. These are `.lnk` or `.url` files and are special Windows files. It is not possible, or at least is not feasible, to create these files using Java. Instead the mail daemon writes the URL in the message text and, for shortcuts, writes a `file://` URL. Almost all mail clients will recognize these URLs and convert them into clickable links.

So, in effect, the feature works the same but just looks different. If you rely on sending shortcuts or URLs in your message, test this make sure it works for your mail clients.

### Documents

When a document is to be attached to the message, the program that is sending the message has to be able to physically read that document using the path provided in the link. Normally, this is expressed in a UNC-style path name, such as `\\Server\Share\Dir\Filename.ext`. The Windows program can just follow this same path to the file in order to attach it. For the IBM i to access that same file located on a Windows server, a number of things must happen:

- You must configure the IBM i computer to enable it to see your Windows network in the IFS. This is a special file system called `/QNTC`. You configure it by configuring and running a service on the IBM i computer called NetServer. This is the same service that you use so that the IBM i computer appears in Network Neighborhood to a PC.
- The mail daemon automatically converts the path name in the task link so that it looks like this `/QNTC/Server/Share/Dir/Filename.ext`, and tries to read that file.
- The user ID on the IBM i computer that the mail daemon is running with must be a valid user name on your Windows network and their passwords must match. There is a global default in TURNOVER® Lifecycle Manager called the “QNTC Adopted authority user”. The mail daemon runs as whatever profile is entered in this default.
- If your tasks links are referenced with drive mappings, the document will not be found and will not be attached to the message.

- The QNTC file system can be unreliable and cannot be “fixed” without an IPL. If the file system is not available, documents attachments will not work. You need to evaluate how important that feature is to you.
- If for any of the above reasons there is a problem accessing the document, then the mail daemon converts the link into a shortcut as a `file://` URL. This may be an acceptable workaround.

You can set a global property to convert all document attachments to shortcuts and not even try to attach them as links. You might want to do this to ensure that large documents are not being emailed around. In general, a shortcut link is more efficient as long as the mail recipients are on your local network and can access the files by the shortcut.

---

## Installing and configuring the mail daemon

The TURNOVER® Lifecycle Manager mail daemon is installed as part of TURNOVER® Lifecycle Manager. You must to configure it and set it up to run as an autostart job in one of your subsystems.

### To configuring the Mail Daemon

- 1 Create and edit a plain-text properties file in the IFS.

You can do this using the EDTF command from a 5250 session, or by connecting to the IFS from a Windows PC and editing the file in Notepad.

- 2 Rename the files in the `/TurnOver/tmaild` folder with the `.bak` extension to remove the extension. You should have the following files:

```
commons-logging.properties
simplelog.properties
template.html
tmaild.properties
turnoverjoblog.properties
```

- 3 Edit the `tmaild.properties` file (see below), and then save the changes.
- 4 Stop the mail daemon, and then restart it.

### Editing the `tmaild.properties` file

This file contains the information that the mail daemon needs to connect to your TURNOVER® Lifecycle Manager environment running on your IBM i computer, and other settings that control its behavior.

## Libraries

The first part of the file contains the information that Mail Daemon needs to connect to TURNOVER® Lifecycle Manager on your IBM i server.

```
turnoverProgramLibrary=SOFTTURN
turnoverDataLibrary=SOFTTURN
turnoverCommandLibrary=SOFTTURN
turnoverEnvironment=ENGLISH
```

Supply the names of the libraries where you have TURNOVER® Lifecycle Manager installed as well as the name of your environment.

## JavaMail properties

The JavaMail properties are used by the JavaMail APIs, which the mail daemon uses to send the messages.

```
mail.debug=false
mail.smtp.host=mail.yourcompany.com
mail.smtp.port=25
mail.smtp.sendpartial=true
```

Change `mail.smtp.host` to the TCP/IP name or address of your SMTP server, and leave the other values as shipped.

## TURNOVER® Lifecycle Manager mail daemon properties

The TURNOVER® Lifecycle Manager mail daemon properties contain some of the basic settings for how the messages are sent.

```
defaultFromUser=TURNOVER® Lifecycle Manager Mail Messenger
defaultFromAddress=someaddress@yourcompany.com
```

```
alwaysAttachDocumentsAsShortcuts=false
alwaysUseDefaultFromAddress=false
```

```
subject=RE: ##TASK## - ##DESC##
recipientSeparator=,
temp=/TurnOver/tmaild/temp
```

```
defaultFromUser and defaultFromAddress
```

When TURNOVER® Lifecycle Manager sends an email message, it typically obtains the message's "from" information from the user responsible for the message being sent. This is always an IBM i user profile; TURNOVER® Lifecycle Manager has to look up that user's name and email address in the database. When TURNOVER® Lifecycle Manager cannot find a value, it sends empty values with the message. The mail daemon then uses the values you have specified for the `defaultFromUser` and `defaultFromAddress` properties.

```
alwaysUseDefaultFromAddress
```

Set the value of this property to “true” to have the Mail Daemon always use the same value for the emails it sends.

`alwaysAttachDocumentAsShortcuts`

Prevent documents from being attached to messages and to have the mail daemon convert them to file URLs instead. Because the URL is specific to your local network, only users on your local network can use the URL. (In general, you leave this value as “false”.) The reasons you might want to use this feature are:

- Savings bandwidth/storage because your mail system is not sending large document attachments.
- Ensuring that a document attachment never leaves your internal network. While the user will still see the URL in the message (including information about the server, share, and document name), they will not be able to access it.

#### **subject**

Specify the format of the message subjects. The two substitution values that you can put in your value are ##TASK## (for the task ID) and ##DESC## (for the task description).

`recipientSeparator`

You can specify, when using standard email addresses, multiple recipients in a single email message. When defining a message, such as an escalation table message, separate the email addresses with a character like a comma or semi-colon.

The default separator is a comma.

To use a different separator, specify it here.

If you use values (specifically \*RSRC, \*RQST, \*COORD, or \*ENTERED) to specify email recipients, you cannot use a character-separated list: you can specify only one value for each message.

`temp`

A path on your IBM i computer where the mail daemon will create temporary files. (The mail daemon uses the temp folder as a place where it can create task link file attachments.) The mail daemon creates the file, attaches it to the message, sends the message, and deletes the file. You should not need to change this value.

#### **Settings for TURNOVER® Helpdesk**

*If you are using TURNOVER® Helpdesk:* When the mail daemon needs to create some kind of TURNOVER® Lifecycle Manager link, instead of creating the link file that is used by the TURNOVER® Client, you can create a URL to the equivalent function in TURNOVER® Helpdesk.

`CreateURLExternal=true`

`createURLInternal=false`

`internalDomain=yourcompany.com`

`changeTaskURL=http://www.yourcompany.com/xdesk/changetask.do?taskID=##TASK#`

```
#
viewTaskURL=http://www.yourcompany.com/xdesk/viewtask.do?taskID=##TASK##
reviewTaskURL=http://www.yourcompany.com/xdesk/changetask.do?taskID=##TASK#
#
formsURL=
changeRequesterURL=
viewRequesterURL=

createURLExternal and createURLInternal
```

Specify "true" or "false".

```
internalDomain
```

Specify your company domain.

The mail daemon categorizes recipients with this domain as Internal, and recipients with a different domain as External: If any recipients of a message are External, the message is considered External.

#### URL properties

For each property, specify the URL to access its associated function.

You must enter the base URL for your setup of TURNOVER® Helpdesk.

If you leave any URL blank, the mail daemon creates a TURNOVER® Lifecycle Manager style link. Several of the URL settings are included for future expansion and currently do not have an equivalent function in TURNOVER® Helpdesk. You should leave these blank.

#### Plain text mail settings

The mail daemon can send messages as plain text or HTML.

```
delim=\n
linksHeading=\n\nLinks:\n\n
attachmentHeading=

delim
```

Do not change this parameter.

```
linksHeading and attachmentHeading
```

These items are generally located at the end of the message, and when they are present, this just gives you a chance to add some text before them to make them stand out. Include some "\n" line breaks (delimiters) to format it correctly.

## HTML mail settings

```
useHTML=false
HTMLTemplate=/TurnOver/tmaild/template.html
delimHTML=<br>
linksHeadingHTML=<br><br><font face="Verdana, Arial, Helvetica, sans-
serif" size="3" color="red"><b><i>Links:</i></b></font><br><br>
attachmentHeadingHTML=
```

The main property, `useHTML`, controls the default message type.

## IBM i connection properties

You should not need to edit these properties, but they must be in the file.

```
as400System=localhost
as400User=*current
as400Password=*current
jdbcUserName=*current
jdbcPassword=*current
jdbcDriver=com.ibm.as400.access.AS400JDBCdriver
jdbcURL=jdbc\:as400\://localhost;translate binary\=true;date format\=iso
```

## SMTP authentication properties

Some email servers will not allow you to send email using SMTP without authenticating. These properties allow you to provide user and password information required for SMTP authentication:

```
# mail.smtp.auth=true
# mail.smtp.username=
# mail.smtp.password=
```

---

# Starting and ending the mail daemon manually

Ultimately, you will want to configure the mail daemon to start as an autostart job (see *'Determining where the autostart job will run' on page 243*). While you are working out the details of the configuration, you can start and stop it manually.

Start the mail daemon with the `TMAILD` command.

Because this command only runs in a batch job, you have to use the `SBMJOB` command to submit the job to batch. If you use `SBMJOB`, submit the job to a multi-threaded job queue. You should do this because this job will not end until you end it with `ENDJOB` and your job queue will get backed up.

The command takes three parameters:

- path to the `tmaild.properties` file. You can likely leave the default value of `/TurnOver/tmaild/tmaild.properties`.

- path to a log file for error messages. Again, you can probably use the default. This file is usually only needed if you are having major problems getting the Daemon started. Once the Mail Daemon is running, it logs all of its runtime messages into the job log of its job, allowing you to monitor it like any other job.
- the version of Java to use; use the default.

Submit the job to a job queue that allows multiple jobs to run at once. Also, make sure that the library list for the job includes your TURNOVER® Lifecycle Manager libraries. When the job starts, a second job named QJVACMDSRV starts in the same subsystem. This is where the Java code runs, and where all of the work is actually done. This is the job that will have messages logged by the daemon, and is also the job to end when you want to end the daemon. If you end the job \*CNTRLD, it will detect that and end itself cleanly.

## Determining where the autostart job will run

It is recommended that you run the mail daemon in a subsystem that is tuned for batch jobs like QBATCH or QPGMR.

Alternatively, you could run the job in the TSERVER subsystem if you are already using that subsystem to run other TURNOVER® Lifecycle Manager autostart jobs. However, this not the recommended approach as that the subsystem is tuned to favor interactive-type jobs. The jobs also run with an interactive priority. For the most part, the mail daemon does not use a lot of resources and that would probably be acceptable. However, the exception is when the job starts. Java jobs have a fairly high CPU percentage when they first start as the Java classes are loaded and JIT compiled. This only lasts for a minute or less; and if your subsystem will always be started during an IPL or after hours, it will not be a problem. The main point is to consider it, and if necessary reconfigure the subsystem to change the job priority.

You will likely want to configure the mail daemon to start and end automatically when one of your subsystems is started and ended.

## Creating a job description

To configure an autostart job, you must have a job description. The job description configures the initial environment and also specifies the command to run to start the job. Specify the TMAILD command (see *'Starting and ending the mail daemon manually' on page 242*). For example:

```
CRTJOB JOB (QGPL/TMAILD)
      JOBQ (QGPL/TSERVER)
      TEXT ('TURNOVER® Lifecycle Manager Mail Daemon')
      USER (TSERVER)
      RQSDTA ('TMAILD PROPS (''/TurnOver/tmaild/tmaild.prope
               rties'') LOGFILE (''/TurnOver/tmaild/logs/tmai
               ld.log'') ')
      INLLIBL (SOFTTURN SOFTTURN D SOFTTURN E)
      JOBMSGQFL (*WRAP)
```

This example creates a job description named TMAILD in QGPL and configured it to use the TSERVER job queue also in QGPL.

USER (*username*)

Must specify a valid user to be an autostart job.

RQSDTA (*command*)

The TMAILD command string to execute.

INLLIBL (*turnover libs*)

Include the TURNOVER® Lifecycle Manager libraries.

## Adding an autostart job entry

1 Run this command:

```
ADDAJE SBSD (QGPL/TSERVER)
      JOB (TMAILD)
      JOBD (QGPL/TMAILD)
```

In this example, you are:

- Adding the autostart job entry to the TSERVER subsystem.
- Specifying the name you want to give the job (TMAILD).
- Specifying the job description you created in the previous step.

2 Stop the subsystem, and then restart it.

---

## Multiple TURNOVER® Lifecycle Manager environments

The TURNOVER® Lifecycle Manager mail daemon can monitor and process only a single TURNOVER® Lifecycle Manager environment.

If you have multiple environments (that is, several TURNOVER® Lifecycle Manager data libraries) for which you need mail processing, create multiple “properties” files that contain the configuration information.

You then set up multiple autostart jobs. In each autostart job, you execute the TMAILD command with a different set of parameters for the properties and log file, as well as the TURNOVER® Lifecycle Manager library list.

### Creating the TSNDMAPI data queue in data libraries

For messaging to work, you must delete the TSNDMAPI data queue in the SOFTTURN library, and then create one of these queues in each environment’s data library.



The mail daemon polls the TSNDMAPI data queue on the IBM i system, looking for messages to send. The data queue contains the message text, as well as information about the message sender and the intended recipient. The mail daemon uses this information to send the message using your SMTP mail server.

You could write programs that issue the `TMAILMSG` command to write to the data queue, or that will read and process the records in the data queue. The program must check for the version information (see table below) before processing the rest of the record. If a record in the data queue does not begin with this version, your program should ignore that record.

#### Record layout of the TSNDMAPI data queue

| Field description | From | To   | Contents |
|-------------------|------|------|----------|
| Version           | 1    | 10   | 'V1R0M0' |
| (Empty)           | 11   | 11   |          |
| Message           | 12   | 267  |          |
| (Empty)           | 268  | 268  |          |
| Task              | 269  | 278  |          |
| (Empty)           | 279  | 279  |          |
| Task description  | 280  | 359  |          |
| (Empty)           | 360  | 360  |          |
| From user name    | 361  | 391  |          |
| (Empty)           | 392  | 392  |          |
| From company      | 393  | 422  |          |
| (Empty)           | 423  | 423  |          |
| From email        | 424  | 679  |          |
| (Empty)           | 680  | 680  |          |
| To user name      | 681  | 711  |          |
| (Empty)           | 712  | 767  |          |
| To email          | 768  | 1023 |          |

---

## Log settings

The mail daemon uses the Apache commons-logging log framework to log messages.

If the Mail Daemon is working and sending messages, you should see messages like this in the job log:

```
[INFO] Messenger - Message sent to auser@softlanding.com
[INFO] Messenger - Message sent to blake@softlanding.com
[INFO] Messenger - Message sent to elms@softlanding.com
[INFO] Messenger - Message sent to smith@softlanding.com
```

If there are any problems that prevent it from sending a message, they are logged. You can increase the amount of messages being logged by editing the `turnoverjoblog.properties` file.

---

## HTML message support

The mail daemon can send HTML formatted messages.

### HTML template

The HTML template must contain all of the look and feel for the message.

The template must contain `##MSG_CONTENT##` at the point in the HTML where the message text should be inserted. This would also include any URL links that were added to the message as well as the "Links Heading" as defined in the configuration.

To include graphics or to use stylesheets, you must load them on a web server, and then refer to them in the message by using the full URL to the file. In other words, the graphics are not sent with the message.

### Example HTML template

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title></title>
  </head>
  <body bgcolor="#FFFFFF" marginwidth="0" marginheight="0" leftmargin="0"
topmargin="0">
    <table width="100%" cellpadding="0" cellspacing="0" border="0">
      <tr bgcolor="black"><td height="50"><IMG border="0"
src="http://support.softlanding.com/images/expressdesklogo.gif" width="189"
height="37"></td></tr>
      <tr><td><pre><font size="2">
        ##MSG_CONTENT##
      </font></pre></td></tr>
```

```
<tr><td>&nbsp;</td></tr>
</table>
</body>
</html>
```

A message sent with that template would look something like this:



---

## Overriding HTML and plain text configuration settings

You can override the HTML and plain text configuration settings on an individual message basis. You can do this by using the email feature of TURNOVER® Lifecycle Manager that uses a report mask. In the report mask, you can use the following directives:

`##TX`

Use this in positions 1-4, and as the only content of, the first line in the report mask. This instructs the mail daemon to send the message as plain text. If the default was to use HTML, this would override the default.

`##TF (optional template name)`

Use this in positions 1-4, and as the only content of, the first line in the report mask. This instructs the mail daemon to use HTML. You can also include a path to an HTML template. For example:

`##TF /TurnOver/tmaild/template2.html`

So, you can use the `##TF` switch in two ways:

- to override a default plain text setting to use HTML
- to specify a custom template. (You might want your default template to be for your internal emails and to have a custom template for customers.)

# 17 Mail integration specification

This document describes the feature designed to help and enhance the Helpdesk process by handling external email communication to create Helpdesk requests.

---

## Purpose

Email has become one of the most popular methods of moving information from person to person, as well as from software to person. Currently, TURNOVER makes extensive use of emailing throughout the task handling process, as a task moves through various stages until completion. Although internal communication is already streamlined using TURNOVER, external correspondence via email remains a fully manual and time consuming practice.

As requesters email new requests to the Helpdesk, someone must monitor a mailbox for such requests, then create new tasks and copy the pertinent information into the tasks manually. Similarly, as TURNOVER sends updates via email to requesters regarding the progress of their original requests, email responses must be monitored, and again information must be added to the tasks manually. Clearly, automating the process of monitoring the mailbox for new messages, adding information to new or existing tasks, and then notifying the Helpdesk of the events would be advantageous. This is exactly what TURNOVER Mail Integration is designed to do.

---

## Overview

TURNOVER Mail Integration (TINBOXD or TURNOVER inbox daemon) relies on an incoming email server, an outgoing email server and a set of configuration properties. The configuration properties define the email addresses receiving email messages, the email settings for forwarding messages when necessary, and TURNOVER and IBM i connection information.

Email accounts on the incoming email server are configured to be monitored by TINBOXD. For each email received, TINBOXD picks up the message and determines if the email is associated with an existing TURNOVER task. If so, the message is handled as a reply to a TURNOVER generated message regarding an existing issue.

If the task is still open, the message text is appended to the details of the TURNOVER task. Attachments included in the email are saved as task links.

If there is no existing TURNOVER task associated with the email, the message is handled as a new request.

Before a new task can be added, TINBOXD first uses the email sender's address to determine if the sender is an existing TURNOVER requester. If so, a new task is created with the sender as the task requester, the email subject as the task description, and the message text as the task details. Attachments included in the email are saved as task links.

The two scenarios described above are considered the mainstream cases for TINBOXD. Cases falling outside of these conditions are, for security reasons, forwarded for manual handling. For instance, a task Id is associated with an email, but the task does not actually exist; or a reply is sent to a task that has been closed out; or the email sender is unknown to TURNOVER.

Email forwarding can be divided into multiple directions. Emails received with unknown senders can be forwarded to one address. Emails replying to closed tasks can be passed on to another address. A third address can be set for messages forwarded as a result of handling errors, such as authorization issues or message parsing problems.

---

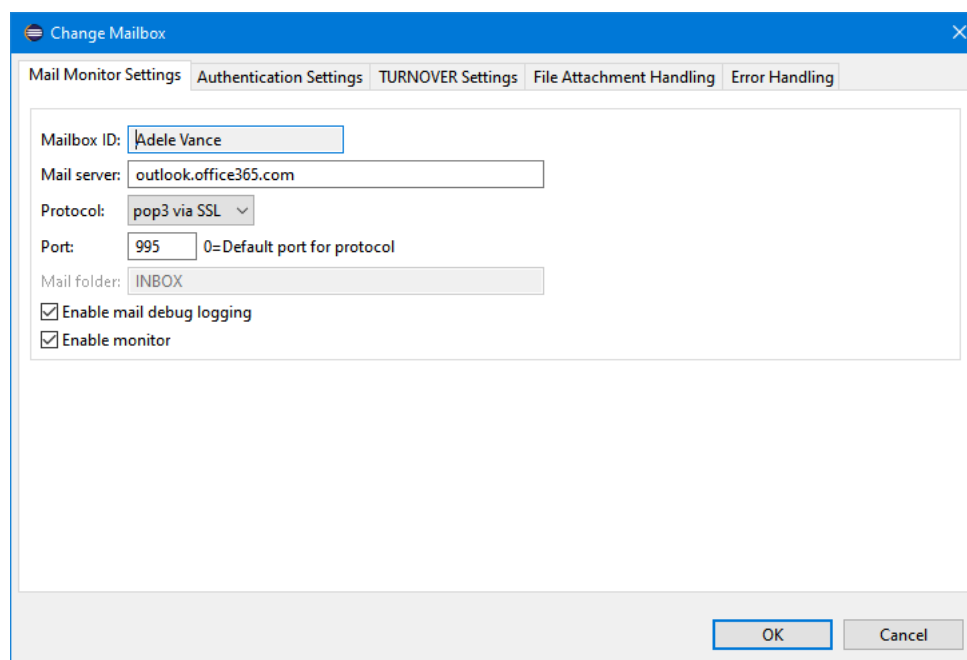
## Configuration properties

Mailboxes to be monitored can be defined in two ways. The TURNOVER client application or RDi plugins can be used to define the properties of one or more mailboxes to be monitored. The mailbox definitions can be found in the Project Administration -> Helpdesk subsystem.

The dialogs for defining mailboxes have been amended to accommodate alternative authentication schemes such as OAuth2. The new panels are as follows:

---

## Mail Monitor settings



**Change Mailbox**

Mail Monitor Settings | Authentication Settings | TURNOVER Settings | File Attachment Handling | Error Handling

Mailbox ID: Adele Vance

Mail server: outlook.office365.com

Protocol: pop3 via SSL

Port: 995 0= Default port for protocol

Mail folder: INBOX

☒ Enable mail debug logging

☒ Enable monitor

OK Cancel

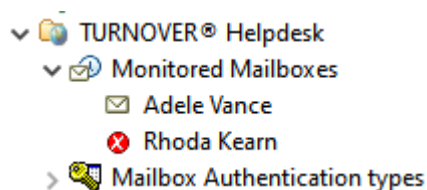
The mailbox ID is a unique name for the mailbox definition and would normally be the mailbox owner's name.

The Mail server, protocol and port must be set to the host name, protocol and port required to access the mail server.

If IMAP protocol is selected then the name of a mailbox folder to be read must be specified. For POP protocol no folder is required. If the IMAP protocol is not selected then this input box will be disabled.

Check the 'Enable mail debug logging' box if you want to see increased log output to help with resolving any issues.

Check the 'Enable monitor' box to enable the mailbox to be monitored. Uncheck this box to prevent the mailbox from being monitored temporarily. The mailbox definition will be marked with a cross to show that it is currently disabled. The figure below illustrates two mailboxes that have been disabled temporarily.



## Authentication settings

A screenshot of a 'Change Mailbox' dialog box. The dialog has a blue title bar and several tabs: 'Mail Monitor Settings', 'Authentication Settings' (which is selected), 'TURNOVER Settings', 'File Attachment Handling', and 'Error Handling'. In the 'Authentication Settings' tab, there is a section for authentication. It includes a checkbox 'Authentication required:' which is checked. Below it is a dropdown menu for 'Authentication Type:' with the selected value 'POP3S using authorisation flow (OUTLOOK\_AUTH\_POP3S)'. Underneath the dropdown are two text input fields: 'Username:' containing 'AdeleV@zf30r.onmicrosoft.com' and 'Password:' containing a series of asterisks. At the bottom right of the dialog are 'OK' and 'Cancel' buttons.

Check the 'Authentication required' box if your mail service provider requires you to log in to the mail server. Depending on the authentication type selected a user name and/or a password may be required.

Select the authentication type that should be used when authenticating a user on this mail server. The currently defined authentication types (see image below) are populated in the drop down list.

Depending on the authentication type selected, a password may or may not be required. The authentication type property 'password\_required' determines whether a password is required on this panel.

## TURNOVER settings

This tab determines the specific TURNOVER properties to be applied to tasks created from monitored mail.

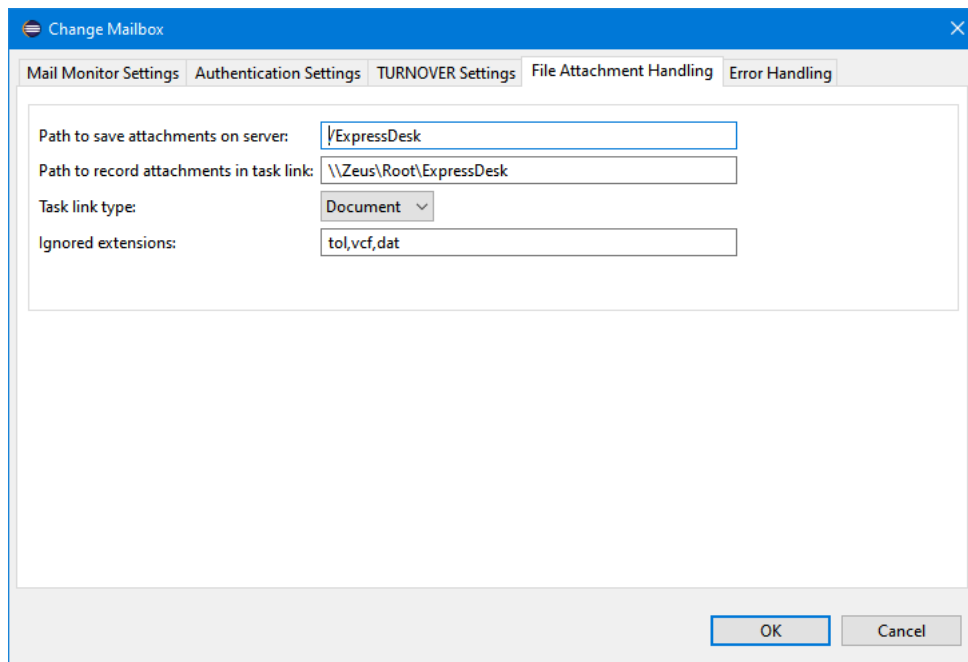
The project field determines the default project that new task will be added to if no specific project is specified.

The TURNOVER user determines the user that will be used as the task requester if the email address of the received email is not an already registered requester.

The regular expression input field can be used to set a mask to match the email subject with a specific project and task that will be used to identify an existing task for update. The regular expression should identify two match groups, the first will be the project identifier and the second the task and optional sub-task identifiers. The default expression will match a subject line in the format "RE:PROJ-001000".

The exit program name and library that will be called when task content is appended. The program is invoked on the IBM i and passed parameters to identify the project, task and sub-task that were amended and the user that amended them. The exit program can perform any tasks the user requires with this information.

## File attachment handling



The screenshot shows a Windows-style dialog box titled 'Change Mailbox'. It has five tabs: 'Mail Monitor Settings', 'Authentication Settings', 'TURNOVER Settings', 'File Attachment Handling' (which is selected), and 'Error Handling'. The 'File Attachment Handling' tab contains four input fields: 'Path to save attachments on server:' with the value '/ExpressDesk', 'Path to record attachments in task link:' with the value '\\Zeus\\Root\\ExpressDesk', 'Task link type:' with a dropdown menu showing 'Document', and 'Ignored extensions:' with the value 'tol,vcf,dat'. At the bottom right of the dialog are 'OK' and 'Cancel' buttons.

This tab controls where files attached to emails will be stored and what type of link will be used to reference the attachments.

The 'Path to save attachments on server' field should refer to a folder in the IBM i IFS that will be the folder to receive the attached files.

The 'Path to record attachments in task link' field should contain a UNC style path that will allow non IBM i clients to access the same folder location. This may also be referred to as the 'virtual path'.

The 'Task link type' determines how the document will be referenced. Either directly as a file name or as a symbolic link.

The 'Ignore extensions' field can be used to tell the monitor to ignore certain attachment file types. These may include image files, arbitrary data files, vcf address files, etc. Any type of file that you want to ignore and not add to the task should be specified in a comma separated list here.



## Error handling

The screenshot shows the 'Change Mailbox' dialog box with the 'Error Handling' tab selected. The dialog has a blue title bar and a close button. Below the title bar are five tabs: 'Mail Monitor Settings', 'Authentication Settings', 'TURNOVER Settings', 'File Attachment Handling', and 'Error Handling'. The 'Error Handling' tab contains the following settings:

- These settings allow you to control who to notify when various problems are encountered.
- Server Settings:
  - SMTP server: smtp.office365.com
  - SMTP port: 587
  - Protocol: smtp (dropdown)
  - Authentication required: ☒
  - Authentication Type: Plain username/password (PASSWORD) (dropdown)
  - Username: AdeleV@zf30r.onmicrosoft.com
  - Password: \*\*\*\*\*
- Forward problems to:
  - Default email address: AdeleV@zf30r.onmicrosoft.com
  - Task is closed: \*DEFAULT (dropdown)
  - No requester found: \*DEFAULT (dropdown)
  - General error: \*DEFAULT (dropdown)

At the bottom right are 'OK' and 'Cancel' buttons.

This tab enables you to control who to notify when various problems are encountered such as an unknown project or task reference, an unknown requester or a requester who is not authorized to a project trying to add a new task or amend an existing task.

The server settings allow you to set the outgoing SMTP mail server host name, port and protocol and whether authentication is required when sending mail with this server.

Many SMTP servers will still allow basic authentication with a simple user and password or require no authentication at all. If necessary you can select an authentication type to be used when logging in to the SMTP server as well as the user and password if required.

The 'Forward problems to' section of this tab allows you to set the email addresses of those people who should be notified in the event of the type of error. You must specify a default email address and you can also specify different email addresses for the specific types of error, or you can set the email address to one of these special values:

| Value    | Description                                                  |
|----------|--------------------------------------------------------------|
| *DEFAULT | The default email address specified here                     |
| *COORD   | The project coordinator as defined in the project properties |
| *RSRC    | The user who is the resource assigned to the task            |
| *RQSTR   | The requester of the original task                           |
| *SENDER  | The sender of the email                                      |

---

## Defining mailboxes in the properties file

Mailboxes can alternatively be configured in a properties file stored in the IFS.

A properties file consists of a set of *key=value* properties that define the various attributes of the mailboxes to be monitored. If the TINBOXD job is to be run on a server other than the one where TURNOVER is installed then a set of properties that describe the connection properties to the installation server are also required.

Mailbox properties are prefixed with a unique mailbox number. Previous versions of TURNOVER had the mailbox number as a suffix. This change has been made to ensure that all of a mailboxes properties are sorted together whenever the properties file is saved. If an old format properties file is used it will be automatically converted to the new format of *n.key=value* instead of *key.n=value*.

Password values that are required to be entered in plain text for either the IBM i server connection, JDBC connection or mailbox user account are encrypted on first use and will be prefixed with the text [ENCRYPTED]. The values will remain encrypted until they are changed.

TINBOXD configuration can be separated into two general categories; the mailbox properties and the connection properties.

The mailbox properties are sets of values prefixed with a property set number associated with each email account for which TINBOXD will be responsible to monitor and handle incoming messages.

The TURNOVER and IBM i connection properties allow TINBOXD to communicate with TURNOVER to perform the necessary task functions. If not explicitly provided the connection properties will default to the localhost, current user and TURNOVER installation in the default library SOFTTURN.

Mailbox properties mirror those already described in the previous section.

### Mailbox Properties

```
0.mailServerDebugLogging=false
0.appendExitLib=APROGRAM
0.appendExitPgm=ALIBRARY
0.authenticate=[false | true]
0.authenticationType=PASSWORD
0.defaultProject=
0.host=<incomingServer.yourCompany.com>
0.id=
0.ignoreAttachmentFileExtensions=
0.linkType=[S | D]
0.mailFolder=INBOX
0.monitorEnabled=[true | false]
0.password=
0.physicalPath=\physicalPath\forAttachmentFiles
0.port=[110 | 995 | 143 | 993]
0.protocolType=[pop3 | pop3s | imap | imaps]
0.regularExpression=
```

```
0.smtp.authenticate=[false | true]
0.smtp.userName=
0.smtp.password=
0.smtp.closedTaskForward=*COORD
0.smtp.defaultForward=anAccount@yourCompany.com
0.smtp.errorsForward=anAccount@yourCompany.com
0.smtp.host=outgoingSever.yourCompany.com
0.smtp.port=25
0.smtp.protocolType=[smtp | smtps]
0.smtp.requesterForward=*COORD
0.smtp.authenticationType=PASSWORD
0.turnoverUser=*CURRENT
0.userName=anAccount@yourCompany.com
0.virtualPath=\\virtualPath\\forTaskLinks
```

#### **TURNOVER Connection Properties**

```
turnoverProgramLibrary=SOFTTURN
turnoverDataLibrary=SOFTTURN
turnoverCommandLibrary=SOFTTURN
turnoverEnvironment=ENGLISH
```

#### **IBM i Connection Properties**

```
as400System=AS400_SYSTEM_NAME
as400User=VALID_USER
as400Password=VALID_PWD
jdbcUserName= VALID_USER
jdbcPassword= VALID_PWD
jdbcDriver=com.ibm.as400.access.AS400JDBCdriver
jdbcURL=jdbc\:as400\://AS400_SYSTEM_NAME;translate binary\=true;date
format\=iso
```

#### **Miscellaneous Properties**

```
secureCurrentUser=true
tokenCacheFilename=./tokens.json
checkMailIntervalSeconds=60
useDbMailboxes=[true | false]
```

New properties have been added to control where the monitor will store cached access tokens for authentication types that operate with tokens (`tokenCacheFilename`). The property `useDbMailboxes` enables you to choose to use the mailboxes as defined by the RDi plugins or standalone client instead of mailboxes defined in the properties file.

Previously, if a properties file was specified when starting the monitor, all mailboxes were expected to be defined in the properties file. Now it may be advantageous to start the monitor job on a server other than the IBM i server but still use the mailbox definitions defined centrally.

---

## Mailbox configuration property descriptions

The TINBOXD mailboxes are based on one email address each. More likely, that email address will correspond to one function in the company, such as external customer support or internal IT operations. Mailboxes properties should be configured with the mailbox function in mind. Configuring the email forwarding properties is an example. Message forwarding requirements will probably differ in the handling of customer support communications coming from outside the company, versus IT operation communications coming from inside the company.

---

## Mailbox properties

Multiple mailboxes can be monitored, so mailbox property names must adhere to specific rules. All mailbox property names must be prefixed with a unique sequential number and a dot (".") and for each group of mailbox settings. Property names for the first mailbox listed must have a prefix of "0."; property names for the second mailbox must have a suffix of "1."; and so on. All mailbox property values are required, unless explicitly specified. A maximum of 100 mailboxes can be defined numbered 0-99 consecutively without any breaks. If a break in sequence occurs processing of mailboxes will stop. If you define mailboxes prefixed '0...', '1...', '3...' then only mailboxes 0 and 1 will be monitored.

The `host`, `port` and `protocolType` properties are the connection settings for the incoming mail server. The port defaults to 110. The `protocolType` defaults to POP3, and can accept the following values:

**POP3** – Post Office Protocol version 3 is the most common method of receiving Internet mail. It supports only one mailbox folder per user account on the server. POP3 has minimum capabilities for message management, deferring the responsibility to email client software. The port used for POP3 is typically 110.

**POP3S** – This is the same as POP3 with the addition of running across a secure connection, or SSL. The SSL port used for this protocol is typically 995.

**IMAP** – Internet Message Access Protocol is a more advanced protocol than POP3, however it can be used only on mail servers that specifically support the protocol. IMAP allows for multiple mailbox folders per user account as well as mailbox folder sharing between accounts. The port used for IMAP is typically 143.

**IMAPS** – This is the same as IMAP with the addition of running across a secure connection, or SSL. The SSL port used for this protocol is typically 993.

The `authenticate` property is true if the server requires authentication, and false if not. The default is true.

The `authenticationType` property should be set to an appropriate authentication type as defined in the TURNOVER client that supports the authentication mechanism required by the server. The default value of PASSWORD should be set for mail servers that authenticate with a basic user and password.

Other mail servers may require more secure authentication types such as OAUTH. TURNOVER is supplied with some sample authentication types for Microsoft Outlook and Google Gmail authentication using various OAuth authentication flows.

The `userName` and `password` properties are required for mail servers requiring basic authentication (`authenticate=true` and `authenticationType=PASSWORD`). These values must equate to a valid account on the server. Some more secure authentication methods may require a password while others may not depending on the authentication type being used.

The `mailFolder` property is the mail folder in which new messages will arrive. This value is not case-sensitive. For POP3, the value can only be INBOX, and this is the default for the property. For IMAP, the value can be INBOX or another folder name. If the folder is not the default root folder for the message store, the value must include the full folder name, including subfolders, relative to the root.

The `turnoverUser` property is not required. In determining TURNOVER authorities for processing emails into tasks, the `as400User` property in the IBM i Connection properties (explained below) is used as the default. The `turnoverUser` property provides an option to use a different profile. A `turnoverUser` value of `*CURRENT` will default back to the `as400User` property value.

The `defaultProject` is the project to which new tasks will be added. The project must be a valid TURNOVER project, and the user specified for the `turnoverUser` property must be authorized to this project.

The `physicalPath` property is the path where the server saves the attachment document. If this is running on an IBM i, you must specify an IBM i IFS path name. The server will create subdirectories in the location specified and then save the attached documents there.

The `virtualPath` property is the same location as the `physicalPath` value, as it needs to be recorded in the database. This should always be in a Windows-style UNC format, and you must enter 2 slashes `\\` for every 1 slash `\` in the path.

The `smtp.host` and `smtp.port` properties are the connection settings for the outgoing mail server used for forwarding messages. The `smtp.port` defaults to 25. An SMTP server requiring a secure connection is now supported.

The `smtp.authenticate` property is true if the server requires authentication, and false if not. The default is true. More secure authentication methods are also available for SMTP servers.

The `smtp.authenticationType` property should be set to an appropriate authentication type as defined in the TURNOVER client that supports the SMTP authentication mechanism required by the server. The default value of `PASSWORD` should be set for mail servers that authenticate with a basic user and password.

The `smtp.userName` and `smtp.password` properties are required for mail servers requiring authentication (`smtp.authenticate = true`). These values must equate to a valid account on the SMTP server. Some more secure authentication methods may require a password while others do not.

The `smtp.closedTaskForward`, `smtp.requesterForward`, and `smtp.errorsForward` properties are for forwarding addresses in cases when the messages cannot be processed due to, respectively, the task being closed, the requester being unknown, or an unexpected error occurring. Each of these properties support a single email address, or one of the following special values:

- \*COORD - The coordinator of the project to which the message is associated.
- \*RSRC - The resource of the task to which the message is associated.
- \*RQSTR - The requester of the task to which the message is associated.
- \*SENDER - The person sending the email message.

The `smtp.defaultForward` property supports only a single email address, and is used as a back up forwarding address in case one of the other forwarding address properties are blank, or in case an error occurs in deciphering one of the other forwarding address properties. If the `smtp.defaultForward` property value is invalid or left blank, messages could ultimately be lost.

The `mailServerDebugLogging` property is true or false. A true value will enable informational logging for email server event processes. The default is false.

The `ignoreAttachmentFileExtensions` property is a comma-delimited list of file extension values. Attached email files containing an extension from the list will be ignored when processing attachments into task links.

The `appendExitPgm` and `appendExitLib` properties define the program to call after appending task details.

The `checkMailIntervalSeconds` is a global property that governs how frequently all mailboxes are checked for new mail messages. The units of this property value are in seconds.

The `secureCurrentUser` property if set to true ensures that the current user of the job running the monitor is used for all connections.

The `tokenCacheFilename` property determines where the TINBOXD job will store cached access tokens when using more secure authentication methods such as OAuth. All tokens cached into this file are encrypted for additional security. This file can be placed anywhere in the IFS but it is recommended that the default location of `./tokens.json` is used to store the token cache in the same folder as the installed software.

The `useDbMailboxes` property can be set to true if the mailbox definitions defined via the TURNOVER client and stored on the TURNOVER server are to be used instead of using mailbox definitions included in the properties file. If the TINBOXD monitor job is run on a server other than the one where TURNOVER is installed you may still want to use the centrally configured mailbox definitions and not have to define them again in this properties file.

---

## TURNOVER connection properties

The `turnoverProgramLibrary`, `turnoverDataLibrary`, `turnoverCommandLibrary`, and `turnoverEnvironment` properties define the libraries where TURNOVER is installed and the TURNOVER environment to be used. If not specified these will default to SOFTTURN, SOFTTURN.D and SOFTTURN.E and environment ENGLISH.

## IBM i connection properties

The `as400System` property is the TCP/IP name or address of the IBM i server. This value can be 'localhost' if the TINBOXD is running on the same IBM i server as TURNOVER.

The `as400User` and `as400Password` property values must be a valid user account in TURNOVER. This account will be used for performing TURNOVER task functions if the `turnoverUser` property in any of the mailbox property groups is *\*CURRENT* or blank. The password can be provided in plain text. The monitor will encrypt the password on first use.

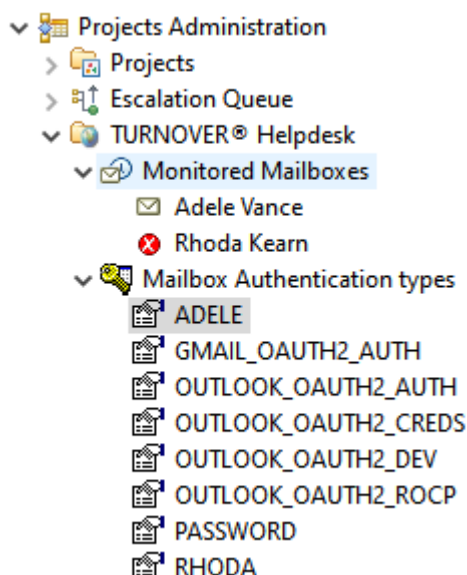
The `jdbcUserName`, `jdbcPassword`, `jdbcDriver`, and `jdbcURL` properties are needed for connecting to the database. The password can be provided in plain text. The monitor will encrypt the password on first use.

## Mail server authentication types

Mail server authentication is becoming increasingly more secure and instead of just a user and password more secure methods of authentication are now being demanded.

To meet these requirements TURNOVER supports the creation of specific authentication types that can be associated with mailboxes to support these new authentication methods.

A new item has been added to the Project Administration subsystem of the TURNOVER client plugins for RDi and the standalone client application named 'Mailbox Authentication Types' that enables specific properties required to support any potential current or future authentication mechanism to be defined.



An authentication type is essentially a specific Java class that supports the authentication protocol and a set of properties that are required by the protocol to effect and authenticate. Currently TURNOVER supports the OAuth2 specification, namely the resource owner credentials, device code, authorization code and client credentials flows supported by mail servers that authenticate using OAuth2 such as Microsoft Office365 and Google Mail (Gmail).

Predefined authentication types for Microsoft Office365 (Outlook) and Google Gmail are provided but require a certain degree of user customization by an administrator that is familiar with Microsoft Azure administration or Google Cloud administration in order to establish suitable enterprise applications that can be configured with the required permissions that will allow authentication of a user using an OAUTH2 access token issued by Microsoft or Google authentication servers in place of a traditional password.

The following authentication types are provided out of the box:

| Property file              | Usage                                                                  |
|----------------------------|------------------------------------------------------------------------|
| GMAIL_OAUTH2_AUTH.props    | Google Mail (Gmail) OAuth2 authorization code flow                     |
| GOOGLE_SERVICE_IMAP.props  | Google Mail (Gmail) OAuth2 service account IMAP                        |
| GOOGLE_SERVICE_SMTP.props  | Google Mail (Gmail) OAuth2 service account SMTP                        |
| OUTLOOK_OAUTH2_AUTH.props  | MS Office365 (Outlook) OAuth2 authorization code flow                  |
| OUTLOOK_OAUTH2_CREDS.props | MS Office365 (Outlook) OAuth2 client credentials flow                  |
| OUTLOOK_OAUTH2_DEV.props   | MS Office365 (Outlook) OAuth2 device code flow                         |
| OUTLOOK_OAUTH2_ROPC.props  | MS Office365 (Outlook) OAuth2 resource owner password credentials flow |
| OUTLOOK_OAUTH2_SMTP.props  | MS Office365 (Outlook) OAuth2 client credentials flow SMTP             |
| PASSWORD.props             | Simple user and password authentication                                |

These properties files are shipped in the folder `"/turnover/tinboxd/config"`. If you have a shared folder mapped to the IBM i server then these standard property sets can be imported when creating a new authentication type using the 'Import' button on the new or change authentication type dialog.

Various properties such as the Azure tenant identifier, application client identifier and client secret values will need to be inserted into these standard templates as described for each authentication flow type below.

Which of these authentication base types you use depends on your email service provider and which of the OAuth or other authentication methods they support.

You can also create your own authentication types and implement your own Java class to support that authentication method as new schemes arise.



---

## Open Authentication Specification v2.0

The Open Authentication Authorization Framework version 2.0 otherwise known as OAuth2 is an authentication mechanism that allows users and services to authenticate without openly exchanging passwords.

The full description and specification of the protocol is described in the internet request for comments (RFC) documents here:

*The OAuth 2.0 Authorization Framework*

*OAuth 2.0 Device Authorization Grant*

Any source of information that requires a user to log in to access the resource can implement a method of authentication that follows this agreed specification. If all providers implement the same protocol then interoperability between resources and users is standardized.

OAuth was primarily designed to facilitate authentication of users to web based services but alternate flow types enable all kinds of other authentication patterns to be used by other services that may not have a direct web presence.

OAuth2 essentially relies on a separate authentication server verifying that a user is who they say they are and that they have been granted permission to access the resource they are requesting access to. The authentication server then issues the user with an access token that they can use to request access to the resource. The access token takes the place of the password in many situations.

Authentication is performed against a real or notional "application" that needs to be defined to the authentication server and that acts as a point of reference and a placeholder for the permissions granted to access the application.

The following authentication flows described in the referenced RFCs are implemented by TINBOXD:

### Resource owner password credentials

*Resource owner password credentials* (ROPC) is an OAUTH2 authentication flow that basically enables authentication to be performed using the simple user and password paradigm but is considered an insecure method of authentication. It may provide a useful stop gap until a more secure alternative can be put in place. The next specification of OAuth after version 2 will very likely remove this authentication flow from the recognized protocol so it should not be considered a long term solution.

### Device code

*Device code* is an OAUTH2 authentication flow that requires the user to enter a code into a form presented on a web page on any device such as a PC or mobile device in order to authenticate the user and to issue an access token to be used as a password.

The first time that a mailbox that requires authentication via the device code flow is logged in a message will be presented to the operator either on the IBM i system operator's message queue or a pop up dialogue if running the monitor job on a server with desktop capabilities.

The message will give a URL of a web page (if running on a desktop enabled server the web page will automatically open) and provide a code. The user is required to open the web page (if not already open) by copying and pasting the URL into a browser and enter the code.

There may be a few more panels displayed on the web page that indicates the application name and what access is being authenticated depending on the authentication server. This will normally be granting permission to read the content of the user mailbox.

Once the device code has been supplied to the web site an access token will be granted to the TINBOXD application that will be stored in the tokens cache file.

Access tokens can be reused to authenticate the user repeatedly but have a short life of up to 90 minutes.

Whenever possible a refresh token may also be supplied by the authentication server. This refresh token enables the application to acquire a new access token before the current one expires which means there is no need to repeat the device code authentication step.

Refresh tokens do have an expiry period but they are much longer lived than access tokens. A refresh token may remain valid for up to three years after which time the customer will be required to repeat the device code authentication step to acquire fresh tokens.

## **Authorization code**

*Authorization code* authentication flow is designed specifically for web applications to authenticate a user. Since the TINBOXD mailbox monitor application is not intended to be a web application this authentication flow is not ideal since it requires the provision of a callback URI that the authentication server will invoke to pass the necessary authentication code back to the application.

If TINBOXD is being run on a desktop enabled server such as Windows then this authentication flow can be used. The application will start a lightweight web server to respond to the authentication redirect response to obtain the necessary code to acquire an access token.

Some providers do provide a way for command line invoked (CLI) applications with no web browser interface to use the authorization code flow such as Google Gmail. With Gmail a special redirect URL can be provided that behaves much like the device code flow previously described.

The authentication request results in the authentication server providing the user with a code that needs to be entered into a web page to complete the authentication phase. Once this has been conducted once the same mechanism of refresh tokens can be used to refresh the access tokens so that the customer does not need to repeat this step again.

## Client credentials

The *client credentials* authentication flow is the one primarily intended to enable background or daemon processes to be authenticated with OAuth but this authentication flow requires much more configuration by the authentication server administrator.

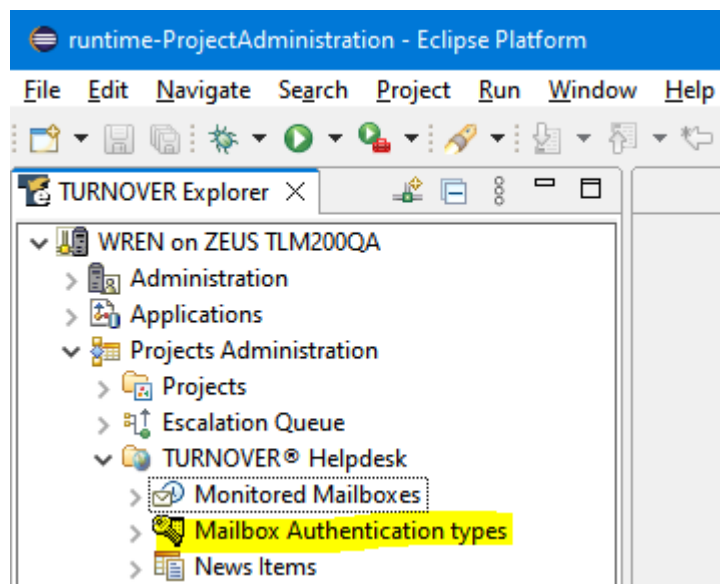
As well as defining an Enterprise application to be granted permissions there also needs to be a service principal account associated with the application that is assigned the necessary permissions. Each authentication server has a different method of defining a service principal account and granting the necessary authorities.

## TURNOVER authentication types

To allow TURNOVER to support this variety of possible authentication flows it is possible to define authentication types in the TURNOVER client application that are essentially a set of properties and a reference to a Java class that will perform the necessary authentication requests in order to acquire an access token.

TURNOVER comes supplied with classes that will follow the four authentication flows previously described. Each requires a set of properties to drive the authentication process.

Authentication types can be defined in the Project Administration -> TURNOVER Helpdesk section of the client application here:



An authentication type is identified by a name and a description and is essentially a set of key/value properties. Each property name must have a value or the authentication type cannot be saved.

As a bare minimum the authentication type must have a 'class\_name' property that identifies the Java class responsible for completing the authentication flow. By default this will be the basic user and password authentication class

'com.softlanding.turnover.mail.authentication.MailboxAuthenticator'. The authentication type named PASSWORD that refers to this class is a fixed entry that cannot be deleted.

The authentication type property set can also contain properties that may be required by the mail server to establish communication such as whether or not to start TLS/SSL for a secure connection or whether to generate debug logging. Any property that starts mail.\${protocol} (where \${protocol} is the mail protocol either POP or IMAP) will be added to the connection properties for the mail session. These additional properties that may be added are described in the Jakarta mail library documentation here:

### *Jakarta Mail Specifications*

If there are additional properties that may be required that are not prefixed with mail.\${protocol} or you run out of available entries in the authentication type property set then they can be added in a property named 'additional\_properties' as a comma separated list of key=value pairs. This property must contain a valid set of key=value pairs before the authentication type can be saved.

If the property contains sensitive information such as passwords or secrets then check the 'Encrypt' box next to the property line. When the property set is saved the value will be securely encrypted and the encrypted value prefixed with the string '[ENCRYPTED]' to indicate that encryption has been applied. Once encrypted the value cannot be decrypted externally. To change the value simply replace it with a new unencrypted value.

The following authentication type implements the resource owner password credentials flow for Microsoft Office365 (Outlook):

| Encrypt                  | Property name:                                             | Property Value:                                                                                |
|--------------------------|------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| <input type="checkbox"/> | class_name                                                 | com.softlanding.turnover.mail.authentication.ResourceOwnerCredentialsAuthenticator             |
| <input type="checkbox"/> | client_id                                                  | <paste Azure AD app client id here>                                                            |
| <input type="checkbox"/> | client_secret                                              | <paste Azure AD app client secret here and encrypt>                                            |
| <input type="checkbox"/> | device_code_uri                                            | https://login.microsoftonline.com/\${tenant}/oauth2/v2.0/devicecode                            |
| <input type="checkbox"/> | grant_type                                                 | password                                                                                       |
| <input type="checkbox"/> | include_client_secret                                      | true                                                                                           |
| <input type="checkbox"/> | mail.\${protocol}.auth.login.disable                       | true                                                                                           |
| <input type="checkbox"/> | mail.\${protocol}.auth.mechanisms                          | XOAUTH2                                                                                        |
| <input type="checkbox"/> | mail.\${protocol}.auth.plain.disable                       | true                                                                                           |
| <input type="checkbox"/> | mail.\${protocol}.auth.xoauth2.disable                     | false                                                                                          |
| <input type="checkbox"/> | mail.\${protocol}.auth.xoauth2.two.line.authentication.for | true                                                                                           |
| <input type="checkbox"/> | password_required                                          | true                                                                                           |
| <input type="checkbox"/> | redirect_uri                                               | https://localhost                                                                              |
| <input type="checkbox"/> | scope                                                      | openid offline_access https://outlook.office.com/IMAP.AccessAsUser.All https://outlook.office. |
| <input type="checkbox"/> | tenant                                                     | <paste Azure AD tenant id here>                                                                |
| <input type="checkbox"/> | token_uri                                                  | https://login.microsoftonline.com/\${tenant}/oauth2/v2.0/token                                 |
| <input type="checkbox"/> |                                                            |                                                                                                |
| <input type="checkbox"/> |                                                            |                                                                                                |
| <input type="checkbox"/> |                                                            |                                                                                                |
| <input type="checkbox"/> |                                                            |                                                                                                |

---

## Common properties

The property name field contains a number of common property names that you may need to add but these are not exclusive. If there are other property names that are required simply enter the name manually. Property names are not case sensitive.

When using the supplied authentication flow classes you must use the property names listed below. For your own authentication classes you can name the properties however you choose.

The common property names supplied are:

| Property name         | Description                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| additional_properties | Add any additional properties as <i>key=value</i> pairs in a comma separated list. For example, <i>prop1=value1, prop2=value2</i> . These properties will be added to the Jakarta mail session properties.                                                                                                                                                                                                                       |
| auth_code_uri         | The authentication server URI to make requests for authorization codes. This value will be supplied in the documentation of the specific authentication server you are using.                                                                                                                                                                                                                                                    |
| app_keys              | For Google mail users authenticated with a service account this property specifies the location of the JSON PPK file that was created for the service account. This file should be located in a secure folder with restricted read only access to the user that runs the mail monitor job to protect the integrity of the service account. If the file is compromised you should create a new PPK file and update this property. |
| class_name            | The name of the Java class that implements the required authentication flow. For more details about the supplied classes see below.                                                                                                                                                                                                                                                                                              |
| client_id             | The authentication server application or client identifier required to identify the application making the request for authentication. This value will be supplied to you by your authentication server administrator responsible for setting up the application.                                                                                                                                                                |
| client_secret         | The client secret that was generated for the application. This value should always be encrypted. This value will be supplied to you by your authentication server administrator responsible for setting up the application.                                                                                                                                                                                                      |
| code                  | Not used by any of the shipped authentication flow classes.                                                                                                                                                                                                                                                                                                                                                                      |

| Property name                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| device_code_uri                             | The authentication server URI to make requests for a device code when using the device code authorization flow. This value will be supplied in the documentation of the specific authentication server you are using.                                                                                                                                                                                                                                                                                                                                                                                                                          |
| empty_trash                                 | For IMAP protocol only. If a trash folder name has been defined (see below) this property determines whether the trash folder should be emptied before closing the session. The folder will be emptied regardless of whether there is any mail to process or not.                                                                                                                                                                                                                                                                                                                                                                              |
| grant_type                                  | The grant type that is being used to authenticate the user. This can be one of password, device_code, authorization_code or client_credentials.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| include_client_secret                       | Set to true if the client_secret value should be included in certain requests to the authentication server                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| password                                    | The mailbox user password if required (see below).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| password_required<br>smtp_password_required | If the authentication type requires a password to be entered for the mailbox user or the SMTP error handling user set these property values to 'true' accordingly. Only the basic authentication and 'password' OAuth grant type require a password. The other OAuth flows authenticate with an access token as the password. If either of these properties are set to 'true' and this authentication type is used by a mailbox account or an SMTP error handling account then you will be required to enter a password in the mailbox authentication settings or error handling dialog tabs.                                                  |
| redirect_uri                                | When using the OAuth authorization code flow the authentication server will redirect the request for an authorization code to the URI specified here. If using the shipped class this redirect URI should be <a href="http://localhost:8080/oauth">http://localhost:8080/oauth</a> . If port 8080 is not available on the server then use a different port number that is available. If running on a desktop enabled server such as Windows the monitor will attempt to start a lightweight web server to accept redirects to this address to retrieve the authorization code and then use that code to request an access token automatically. |

| Property name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| response_type | For OAuth authorization code flows the response type in the request defaults to 'code' but other request types may have a different response type value. When using the shipped authentication classes do not specify this property and it will default to 'code'.                                                                                                                                                                                                                                                                                                                                         |
| state         | Some OAuth authentication flows allow a state value to be provided in each request which is echoed back in any response. You can set a state value here but it will not be used by the shipped authentication flow classes, which use an internally generated UUID value for the state. You may prefer to use a different state for your own authentication classes.                                                                                                                                                                                                                                       |
| tenant        | When authentication with a Microsoft Azure Entra ID server the specific tenant identifier where the application is installed may be required in authentication requests. Enter the tenant id. here.                                                                                                                                                                                                                                                                                                                                                                                                        |
| token_uri     | The authentication server URI to make requests for an access token when using any of the predefined authorization flows.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| trash_folder  | For IMAP protocol only. If the name of a trash folder is specified by this property all handled mail messages will be moved to this folder at the end of the session. Some mail servers such as Google mail (Gmail) do not fully delete a message marked as deleted. They just remove the original label on the message and leave it visible in the 'all mail' folder. Use this property in conjunction with the empty_trash property above to completely delete messages from the server. For Google mail this folder is usually named [gmail]/trash but national language encoding may change this name. |
| username      | The mailbox user name                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

## Supplied authentication type classes

The class\_name property specifies the package name of the Java class that will be invoked to implement the authentication flow. The type of authentication flow will determine what other properties are required.

Five classes are provided as standard to support basic authentication and the four OAuth authentication flows. These are as follows:

```
com.softlanding.turnover.mail.authentication.MailboxAuthenticator  
com.softlanding.turnover.mail.authentication.ResourceOwnerCredentialsAuthen  
ticator  
com.softlanding.turnover.mail.authentication.AuthCodeAuthenticator  
com.softlanding.turnover.mail.authentication.ClientCredentialsAuthenticator  
com.softlanding.turnover.mail.authentication.DeviceCodeAuthenticator
```

To enable easy configuration of an authentication type there are five sample property sets that are shipped in the tinboxd/config folder in the location where you installed TINBOXD either in the IFS or on your Windows server. You can use these property sets as the base for creating your own authentication types:

```
PASSWORD.props - Basic user/password authentication (no OAUTH2)  
OUTLOOK_OAUTH2_AUTH.props - Authorization code authentication  
OUTLOOK_OAUTH2_CREDS.props - Client credentials authentication  
OUTLOOK_OAUTH2_DEV.props - Device code authentication  
OUTLOOK_OAUTH2_ROCP.props - Resource owner credentials authentication  
OUTLOOK_OAUTH2_SMTP.props - SMTP client credentials authentication  
GOOGLE_SERVICE_IMAP - Google service account authenticator for IMAP  
GOOGLE_SERVICE_SMTP - Google service account authenticator for SMTP
```

---

## Roll your own authentication class

If a new authentication flow is added or a different authentication strategy is required to be implemented new classes can be added that extend the base Jakarta mail Authenticator class:

```
javax.mail.Authenticator
```

To have access to the full set of TURNOVER mailbox properties the class should extend the simple password authenticator class:

```
com.softlanding.turnover.mail.authentication.MailboxAuthenticator
```

To implement another server specific version of any of the OAuth2 authentication flows it may be useful, but not necessary, to extend the class:

```
com.softlanding.turnover.mail.authentication.Oauth2Authenticator
```

However the class is implemented in order to make it available to the monitor process the compiled class file must be placed in the same folder as the tinboxd.jar file and be stored in the normal Java package folder structure then referenced in an authentication type as the class\_name property.

For example, if your class is in package com.mycompany.mail.myauthenticator then the compiled class file should be stored in folder structure:



```
/turnover
|- tinboxd
|- com
|- mycompany
|- mail
|- myauthenticator
|-Authenticator.class
```

---

## Substitution variables

The following substitution variables can be used in authentication type property keys and values and will be replaced by the corresponding property value. This is to help prevent the need to declare additional authentication types where just the mail server protocol is different, for example.

| Variable                                                                 | Description                                                                                      |
|--------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|
| \${protocol}                                                             | The mailbox server protocol that is defined in the mailbox mail monitor settings tab.            |
| \${port}                                                                 | The mail server port that is defined in the mailbox mail monitor settings tab.                   |
| \${user}                                                                 | The mailbox user name that is defined in the mailbox mail monitor settings tab.                  |
| \${password}                                                             | The mailbox user password that is defined in the mailbox definition 'Mail Monitor Settings' tab. |
| \${client_id}                                                            | The value specified in the 'client_id' property.                                                 |
| \${client_secret}                                                        | The value specified in the 'client_secret' property.                                             |
| \${scope}                                                                | The value specified in the 'scope' property.                                                     |
| \${tenant}                                                               | The value specified in the 'tenant' property.                                                    |
| \${grant_type}                                                           | The value specified in the 'grant_type' property.                                                |
| \${smtp_protocol}<br>\${smtp_port}<br>\${smtp_user}<br>\${smtp_password} | The same values can be defined for the selected SMTP authentication type.                        |

---

## Authentication server configuration

The next sections will briefly describe the requirements to configure two of the most widely used authentication services, Microsoft Azure Entra ID and Google Cloud. You should contact your organization's administrator of these services to configure them appropriately for TURNOVER mail monitor to support OAuth authentication.

---

## Firewall considerations

As these authentication methods require access to external web services you should ensure that any of the platforms where you run the TINBOXD monitor have the ability to contact the required web servers to effect the authentication requirements. This may mean configuring any firewall rules to ensure that access is available. Any attempt to contact a web service required to authenticate a user or to access the mailbox that results in an error that says the remote server refused the connection is likely to be a problem with your firewall rules.

Configuration of your firewall rules is something a network administrator in your organization would need to address and is beyond the scope of this document but you should bear in mind that large organizations such as Microsoft and Google may have many distinct IP addresses in load balancing pools that may cause problems with your firewall rules and lead to unpredictable results. Please review all firewall rules before raising any support calls.

---

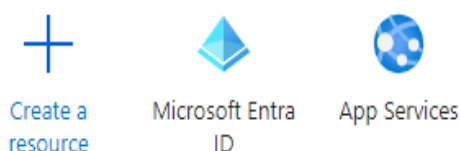
## Microsoft Entra ID Service

To support OAuth authentication on Microsoft Azure Entra ID service you will need to follow these steps to create an application and assign the relevant permissions.

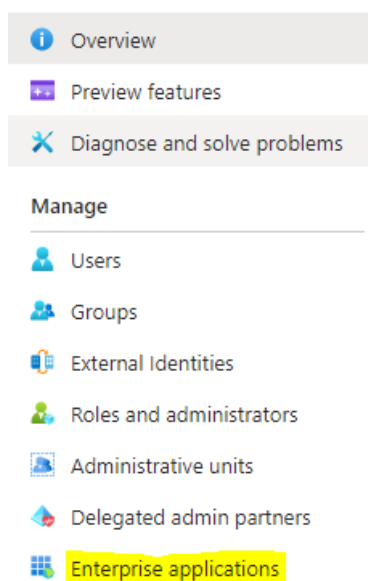
Sign on as an administrator to your Microsoft Azure portal and select the Microsoft Entra ID service:

<https://portal.azure.com/#home>

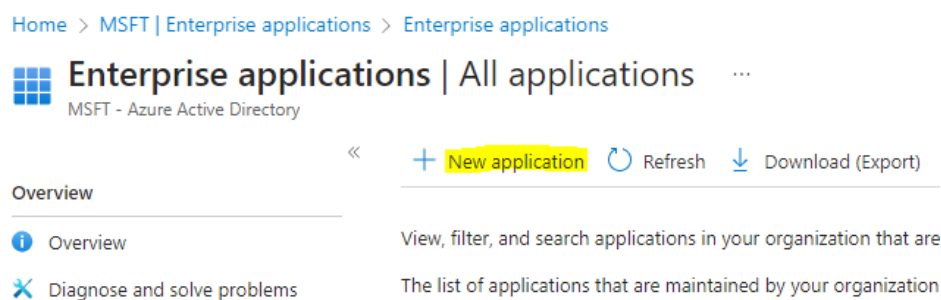
### Azure services



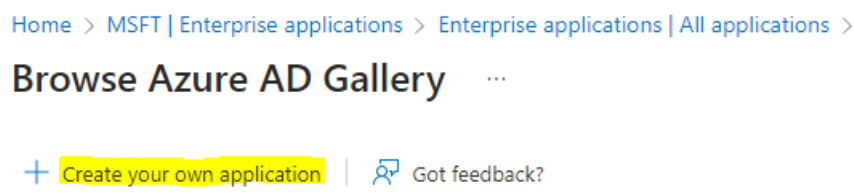
Select Enterprise applications from the navigation bar:



Create a new application:



Create your own application



Register an application to integrate with Azure AD:

## Create your own application



 Got feedback?

If you are developing your own application, using Application Proxy, or want to integrate an application that is not in the gallery, you can create your own application here.

What's the name of your app?

TURNOVER-TINBOXD ✓

What are you looking to do with your application?

- ☐ Configure Application Proxy for secure remote access to an on-premises application
- ☒ Register an application to integrate with Microsoft Entra ID (App you're developing)
- ☐ Integrate any other application you don't find in the gallery (Non-gallery)

We recommend giving the application a name such as SOFTLANDING-TINBOXD or TURNOVER-TINBOXD so that it is easily recognizable.

Choose which accounts you want to have access to this application. We recommend you choose accounts in this organizational directory only (single tenant).

For the redirect URI choose a web platform and enter the URI `https://localhost:8080/oauth` and click the 'Register' button at the foot of the page.

[Home](#) > [MSFT | Enterprise applications](#) > [Enterprise applications | All applications](#) > [Browse Microsoft Entra ID Gallery](#) >

### Register an application ...

\* Name

The user-facing display name for this application (this can be changed later).

TURNOVER-TINBOXD

#### Supported account types

Who can use this application or access this API?

- ☒ Accounts in this organizational directory only (MSFT only - Single tenant)
- ☐ Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant)
- ☐ Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
- ☐ Personal Microsoft accounts only

[Help me choose...](#)

#### Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Web http://localhost:8080/oauth ✓

Return to the Enterprise applications panel and your new application will be listed with any other applications you may have already. A summary of the application information is provided as follows:

3 applications found

| Name                                          | ↑↓ | Object ID                            | Application ID                        |
|-----------------------------------------------|----|--------------------------------------|---------------------------------------|
| <b>TU</b> <a href="#">TURNOVER-TINBOXD</a>    |    | ac4c3d8c-6612-4ca7-9928-eabdb5c42cde | 8a1b0730-0015-45b7-99e1-8d9ab2ce3436  |
| <b>SO</b> <a href="#">SOFTLANDING-TINB...</a> |    | cc2f1d98-986e-47e3-9bd1-bdd226ffecf4 | 60d8dadac-ee94-4901-91a3-f274522d1299 |
| <b>CO</b> <a href="#">ColumbusOM</a>          |    | fe50ff83-4438-49f3-bb1c-07d977bae972 | 5498ce3a-6053-4b61-b9a4-9e7905747249  |

Select your new application to see the properties associated with it:

## Properties

**TU** Name ⓘ  
TURNOVER-TINBOXD ⓘ

Application ID ⓘ  
8a1b0730-0015-45b7-99e1... ⓘ

Object ID ⓘ  
ac4c3d8c-6612-4ca7-9928-... ⓘ

## Getting Started



**1. Assign users and groups**  
Provide specific users and groups access to the applications  
[Assign users and groups](#)

The application (or client) ID is the unique ID of this application in your directory. You can use this application ID if you ever need help from Microsoft Support, or if you want to perform operations against this specific instance of the application using the Azure Graph or PowerShell APIs.

The application (or client) ID should be copied from here and added to your TURNOVER authentication type property set as the 'client\_id' property value.

The Object ID is the unique ID of the service principal object associated with this application. This ID can be useful when performing management operations against this application using PowerShell or other programmatic interfaces.

Click on 'Assign users and groups' to select which users you want to be able to access your application:

+ Add user/group
|
Edit assignment
|
Remove
|
Update credentials
|
Columns
|
Got feedback?

The application will not appear for assigned users within My Apps. Set 'visible to users?' to yes in properties to enable this. →

Assign users and groups to app-roles for your application here. To create new app-roles for this application, use the [application registration](#).

First 200 shown, to search all users & gro...

| Display Name                     | Object Type |
|----------------------------------|-------------|
| No application assignments found |             |

Click on Add user/group and add the user(s) of the mailbox that you want to access:

On the next panel click on the 'None Selected' link to get a list of available users. Select one or more users from the list and click the 'Assign' button.

+ Add user/group
|
Edit assignment
|
Remove
|
Update credentials
|
Columns
|
Got feedback?

The application will not appear for assigned users within My Apps. Set 'visible to users?' to yes in properties to enable this. →

Assign users and groups to app-roles for your application here. To create new app-roles for this application, use the [application registration](#).

First 200 shown, to search all users & gro...

| Display Name                                                                                                              | Object Type |
|---------------------------------------------------------------------------------------------------------------------------|-------------|
| <input type="checkbox"/>  Alex Wilber  | User        |
| <input type="checkbox"/>  Grady Archie | User        |

## Add application owner

You need to add yourself (and any other users) as the application owner so that you can manage the application. Return to the Enterprise applications overview panel and click the application. From the left-hand navigation panel click the 'Owners' link.

+ Add
|
Remove
|
Got feedback?

Assigning owners is a simple way to grant the ability to manage all aspects of an enterprise application. Existing owners can also add or remove other owners. [Learn more](#)

The separate list of owners who maintain the application registration for this app is on the [application registration](#).

Search Owners

| Display Name                | Role assigned |
|-----------------------------|---------------|
| No application owners found |               |

Click on the '+ Add' link and select the users that you want to assign as owners of this application and click 'Select'.

Assigning owners is a simple way to grant the ability to manage all aspects of an enterprise application. Existing owners can also add or remove other owners. [Learn more](#)

The separate list of owners who maintain the application registration for this app is on the [application registration](#).

| Search Owners                                                                                                             |                     |
|---------------------------------------------------------------------------------------------------------------------------|---------------------|
| Display Name                                                                                                              | Role assigned       |
| <input type="checkbox"/>  Dawn Swift     | Configuration Owner |
| <input type="checkbox"/>  Jason Anderson | Configuration Owner |
| <input type="checkbox"/>  Leo Gemma      | Configuration Owner |
| <input type="checkbox"/>  Paul Wren      | Configuration Owner |

You also need to add one or more users as owners who maintain the application registration. Click the application registration link to add application registration owners. Click the 'Add owners' button. Select the users you want to be assigned as registration owners and click 'Select'.

The users listed here can view and edit this application registration. Additionally, any user (may not be listed here) with administrative privileges to manage any application (e.g., Global Administrator, Cloud App Administrator etc.) can view and edit the application registrations. Currently, only individual users are supported as owners of applications. Assignment of groups as owners is not yet supported. If the user setting "Restrict access to Microsoft Entra ID administration portal" is set to Yes, non-admin users will not be able to use the Azure portal to manage the applications they own. [Learn more](#)

| Name                                                                                                                   | Email                           | User name                       | Job Title | Type   |
|------------------------------------------------------------------------------------------------------------------------|---------------------------------|---------------------------------|-----------|--------|
| <input type="checkbox"/>  Paul Wren | paul.wren@xpny0.onmicrosoft.com | paul.wren@xpny0.onmicrosoft.com |           | Member |

## Client secret

You now need to generate a client secret that will be used when negotiating authentication for the user to the application. Return to the Enterprise applications overview panel and click the 'application registrations' link:

[+ New application](#) [Refresh](#) [Download \(Export\)](#) | [Preview info](#) | [Columns](#) | [Preview features](#)

View, filter, and search applications in your organization that are set up to use your Azure AD tenant as their Identity Provider.

The list of applications that are maintained by your organization are in [application registrations](#).

A list of the applications maintained by your organization will be displayed. Click on your application name to view the essential information pertaining to your application registration:

| Essentials              |                                        |
|-------------------------|----------------------------------------|
| Display name            | : <a href="#">SOFTLANDING-TINBOXD</a>  |
| Application (client) ID | : b11507b8-1bb7-47ae-91f9-5e93382dc5d5 |
| Object ID               | : 3346fd62-29de-4d21-b4ac-b521fef43aa0 |
| Directory (tenant) ID   | : f6ed2b9d-305c-4785-888b-11100e92811e |
| Supported account types | : <a href="#">My organization only</a> |

Client credentials : [Add a certificate or secret](#)  
 Redirect URIs : [1 web, 0 spa, 0 public client](#)  
 Application ID URI : [Add an Application ID URI](#)  
 Managed application in I... : [SOFTLANDING-TINBOXD](#)

Click on the 'Add a certificate or secret' link:

Certificates (0) **Client secrets (0)** Federated credentials (0)

A secret string that the application uses to prove its identity when requ

[+ New client secret](#)

| Description                                               | Expires |
|-----------------------------------------------------------|---------|
| No client secrets have been created for this application. |         |

Add a new client secret.

## Add a client secret

Description

Expires

Give the secret a description and select an expiry interval in days. When the secret expires you will need to generate a new secret and authenticate users again with the new secret in place. You must not disclose this secret code and when you store it in the authentication type definition in the TURNOVER client as the 'client\_secret' property you must choose to encrypt the value.

Client secret values cannot be viewed, except for immediately after creation. Be sure to copy and save the secret when created before leaving the page:

Certificates (0) **Client secrets (1)** Federated credentials (0)

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

[+ New client secret](#)

| Description    | Expires   | Value ⓘ                                            | Secret ID                            |
|----------------|-----------|----------------------------------------------------|--------------------------------------|
| TINBOXD secret | 3/13/2025 | <a href="#">~q_8Q~UbcpdYqwOX5IzXpTf6.7bHOZM...</a> | 1d8014d9-318b-4269-9a4c-d0d995479f2d |

## Permissions

You now need to assign the required permissions or scope to the application to enable the TURNOVER monitor to access email mailboxes.

If you are not using client credentials authentication flow continue here. If you are using the client credentials flow you will need to assign alternative permissions to the application and create a service principal account to act on behalf of the application. Continue at section 'Configure Client Credentials Flow'.



## Configure Client Credentials flow

The client credentials flow allows an application to authenticate against permitted resources by using a service principal account that is a trusted application user that the administrator must create and then assign application level permissions to the application.

This authentication flow requires no direct user interaction and is best suited to applications that run as background or daemon processes on servers that do not have graphic UI desktop capabilities or access to an internet web browser.

You should ask your Azure administrator to create a service principal user account for your application using the Windows Powershell interface as follows.

Start a PowerShell session as an administrator.

Install and import the Powershell module for Exchange management if not already installed:

```
Install-Module -Name ExchangeOnlineManagement -Force
Import-module ExchangeOnlineManagement
```

For the next step you will need your tenant id which can be found in the Azure administration console.

Select the 'Microsoft Entra ID' option from the home page and the tenant ID will be displayed in the overview panel. Click the copy button to capture the id to the clipboard.

### Basic information

|                |                                                                                                                            |
|----------------|----------------------------------------------------------------------------------------------------------------------------|
| Name           | MSFT                                                                                                                       |
| Tenant ID      | f8461afa-991e-4fed-bd53-eb3743bf345d  |
| Primary domain | xpry0.onmicrosoft.com                                                                                                      |
| License        | Microsoft Entra ID P2                                                                                                      |

Enter the following PowerShell command to connect to your tenant:

```
Connect-ExchangeOnline -Organization <your.Azure.tenant.id.here>
```

You will be prompted to sign in to your Azure account.

For the next step you will need your application id and the service principal id which can be found in the Azure administration console. From the home panel select 'Microsoft Entra ID' then 'Enterprise Applications'. Select your application from the list and the application id and object id are displayed here. Make sure not to use the Object ID from the Overview of the App Registrations node. Using the incorrect Object ID will cause an authentication failure.

Add a service principal for the application.

```
New-ServicePrincipal -AppId <application id> -ServiceId <object id>
```

Add permission to the service principal to access a specific user mailbox.

```
Add-MailboxPermission -Identity "user@your.org.domain"
                        -User <service principal id>
                        -AccessRights FullAccess
```

Where `-User` is the service principal id, `-Identity` is the user account of the mailbox you want to give access to. This gives full access to the mailbox but not the permission to send mail as the user.

You must now return to the Azure administration console to assign application permissions to your application.

- From the administration console home page select 'Microsoft Entra ID' then 'Enterprise Applications' and click on your application registration.
- Select 'Permissions' from the left hand navigation panel and click on 'app registration' link to set application permissions.

### Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. [Learn more about permissions and consent](#)

[+ Add a permission](#) [✓ Grant admin consent for MSFT](#)

| API / Permissions name                | Type      | Description                   | Admin c |
|---------------------------------------|-----------|-------------------------------|---------|
| <a href="#">▼ Microsoft Graph (1)</a> |           |                               |         |
| <a href="#">User.Read</a>             | Delegated | Sign in and read user profile | No      |

- Click on 'Add a permission' and choose 'APIs my organization uses' then filter the list for 'Office 365 Exchange'.
- From the next panel choose 'Application permissions'.
- Select the permissions IMAP.AccessAsApp, POP.AccessAsApp and SMTP.SendAsApp and click 'Add permissions'.

Now you need to grant admin consent for your organization for these permissions. Click the 'Grant admin consent for...' button and confirm that you want to grant the permissions.

### Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. [Learn more about permissions and consent](#)

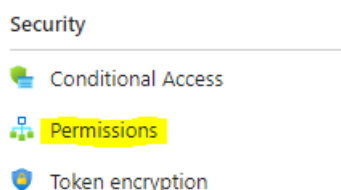
[+ Add a permission](#) [✓ Grant admin consent for MSFT](#)

| API / Permissions name           | Type        | Description                                         | Admin consent requ... | Status             |
|----------------------------------|-------------|-----------------------------------------------------|-----------------------|--------------------|
| ▼ Microsoft Graph (1)            |             |                                                     |                       |                    |
| User.Read                        | Delegated   | Sign in and read user profile                       | No                    | ✓ Granted for MSFT |
| ▼ Office 365 Exchange Online (3) |             |                                                     |                       |                    |
| IMAP.AccessAsApp                 | Application | IMAP.AccessAsApp                                    | Yes                   | ✓ Granted for MSFT |
| POP.AccessAsApp                  | Application | POP.AccessAsApp                                     | Yes                   | ✓ Granted for MSFT |
| SMTP.SendAsApp                   | Application | Application access for sending emails via SMTP AUTH | Yes                   | ✓ Granted for MSFT |

This completes the configuration of client credentials authentication for your application. You now need to set the correct scope property in your TURNOVER mailbox authentication type that will request the correct permissions when logging in to the mail server. Continue are section 'Setting the scope property'.

## Configure Authorization Code Flow

Return to your application overview and click on 'Permissions' in the Security section of the navigation panel:



This panel will show you what permissions the application has been granted and by whom. Some permissions may be delegated to specific users to grant and others may be granted explicitly by an administrator.

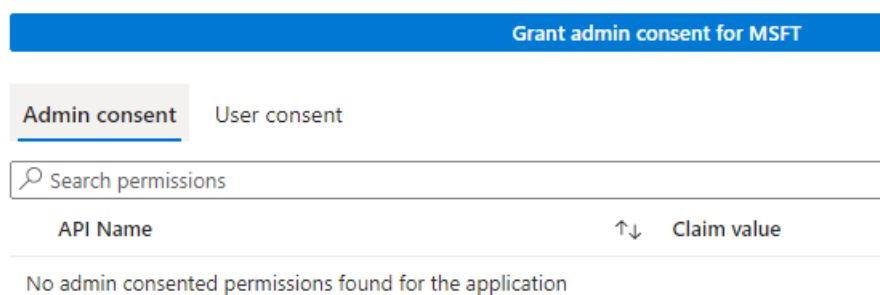
To add permissions to your application click on the 'application registration' link here:

### Permissions

Applications can be granted permissions to your organization and its data by three methods: an to the application. [Learn more.](#)

To request additional permissions for this application, use the [application registration](#).

As an administrator you can grant consent on behalf of all users in this tenant, ensuring that enc



Click 'Add a permission' to add the required permissions:

#### Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. [Learn more about permissions and consent](#)

[+ Add a permission](#) [✓ Grant admin consent for MSFT](#)

| API / Permissions name | Type      | Description                   | Admin consent requ... | Status |
|------------------------|-----------|-------------------------------|-----------------------|--------|
| Microsoft Graph (1)    |           |                               |                       | ...    |
| User.Read              | Delegated | Sign in and read user profile | No                    | ...    |

For resource owner password credentials, device code or authorization code OAuth flows you need to add Microsoft graph permissions as follows:

## Request API permissions

Select an API

Microsoft APIs APIs my organization uses My APIs

Commonly used Microsoft APIs



**Microsoft Graph**

Take advantage of the tremendous amount of data Access Azure AD, Excel, Intune, Outlook/Exchange, single endpoint.

Click Microsoft Graph and choose 'Delegated permissions'. You will need to search for and add the following permissions to your application:

Openid, offline\_access, IMAP.AccessAsUser.All, POP.AccessAsUser.All, SMTP.Send

Add these permissions and your configured permissions should now appear as follows:

#### Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. [Learn more about permissions and consent](#)

[+ Add a permission](#) [✓ Grant admin consent for MSFT](#)

| API / Permissions name | Type      | Description                                         | Admin consent requ... |
|------------------------|-----------|-----------------------------------------------------|-----------------------|
| Microsoft Graph (5)    |           |                                                     |                       |
| IMAP.AccessAsUser.All  | Delegated | Read and write access to mailboxes via IMAP.        | No                    |
| offline_access         | Delegated | Maintain access to data you have given it access to | No                    |
| openid                 | Delegated | Sign users in                                       | No                    |
| POP.AccessAsUser.All   | Delegated | Read and write access to mailboxes via POP.         | No                    |
| User.Read              | Delegated | Sign in and read user profile                       | No                    |

You will need to grant admin. consent for your organization (in this case MSFT) by clicking on the link and confirming the action. The configured permissions will now appear as follows:

#### Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should all the permissions the application needs. [Learn more about permissions and consent](#)

Add a permission

Grant admin consent for MSFT

| API / Permissions name | Type      | Description                                         | Admin consent requ... | Status           |
|------------------------|-----------|-----------------------------------------------------|-----------------------|------------------|
| Microsoft Graph (5)    |           |                                                     |                       |                  |
| IMAP.AccessAsUser.All  | Delegated | Read and write access to mailboxes via IMAP.        | No                    | Granted for MSFT |
| offline_access         | Delegated | Maintain access to data you have given it access to | No                    | Granted for MSFT |
| openid                 | Delegated | Sign users in                                       | No                    | Granted for MSFT |
| POP.AccessAsUser.All   | Delegated | Read and write access to mailboxes via POP.         | No                    | Granted for MSFT |
| User.Read              | Delegated | Sign in and read user profile                       | No                    | Granted for MSFT |

See *'Setting the scope property'* on page 281.

## Setting the scope property

The permissions you have just granted should be mirrored in the TURNOVER authentication type property set in the property named 'scope'.

The scope property determines what permissions you are requesting when you make an authentication request. If the scope you are requesting has not been granted to the application then authentication will fail.

If you have configured client credentials flow authentication then set the scope property to:

`https://outlook.office365.com/.default`

If you have configured authentication flow you must specify the following scopes. You can request fewer permissions than have been granted but you should always specify 'openid' and 'offline\_access' if you want the authentication server to provide refresh tokens that will prevent you from having to repeat the authentication process each time the most recent access token expires which is usually every 90 minutes.

For example; for an authentication type that uses the device code or authorization code flow types and you have set the permissions detailed here then the 'scope' property should have a value of:

openid offline\_access

`https://outlook.office.com/IMAP.AccessAsUser.All`

`https://outlook.office.com/POP.AccessAsUser.All`

`https://outlook.office.com/SMTP.Send`

If you only intend to use the IMAP protocol for this authentication type you can drop the `https://outlook.office.com/POP.AccessAsUser.All` permission from the scope property.

If you do not intend to use OAuth to authenticate access to the SMTP server you can drop the <https://outlook.office.com/SMTP.Send> from the scope property.

For an authentication type that uses the client credentials flow set the 'scope' property to 'https://outlook.office365.com/.default'

---

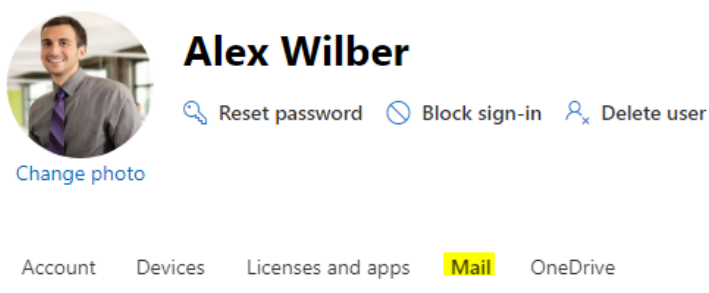
## Enable POP, IMAP and SMTP

You must now enable access to the legacy mail protocols IMAP, POP and SMTP for the Office365 Outlook users whose mailboxes you will be accessing. Failure to perform this step will result in your attempt to log in to the mail server to fail.

Sign in to the Microsoft 365 administration console here:

<https://www.microsoft.com/en-gb/microsoft-365/business/office-365-administration>

Work with active users registered to use Microsoft 365 applications and find the user for whom you want to enable legacy protocol access.



Manage email apps for this user and make sure that the required legacy mail server protocols are checked:

## Manage email apps

Choose the apps where Adele Vance can access Microsoft 365 email.

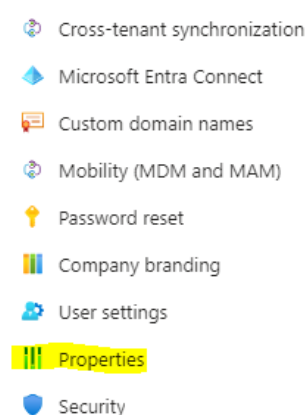
- ☒ Outlook on the web
- ☒ Outlook desktop (MAPI)
- ☒ Exchange web services
- ☒ Mobile (Exchange ActiveSync)
- ☒ IMAP
- ☒ Pop
- ☒ Authenticated SMTP

## Configure SMTP authentication

Microsoft Office365 SMTP server can be used to send email from a configured account. The server log in can be authenticated two ways; basic authentication with just a user account and password or via OAUTH2.

In order to use basic authentication whereby you just need to provide a user account and password it is necessary to disable security defaults in the Azure administration console or you must add the user account to a conditional access rule exempting that account from the security defaults.

To disable security defaults follow these steps. Sign on to the Azure administration console and select the Microsoft Entra ID service then select Properties.



At the bottom of the page select the 'Manage security defaults' link.

### Security defaults

Security defaults are basic identity security mechanisms  
[Learn more](#) 

 Your organization is protected by security defaults.  
[Manage security defaults](#)

In the right hand panel select 'Disabled' in the drop down box.

## Security defaults

×

Security defaults

Disabled

 With security defaults disabled, your organization is vulnerable to common identity-related attacks.

99.9% of account compromise could be stopped by using multifactor authentication, which is a feature that security defaults provides.

Microsoft's security teams see a drop of 80% in compromise rate when security defaults are enabled.

**Reason for disabling \***

This feedback will be used to improve Microsoft products and services. [View privacy statement](#)

☐ My organization is using Conditional Access

☐ My organization is unable to use apps/devices

☐ Too many sign-in multifactor authentication challenges

☐ Too many multifactor authentication sign-up requests

☐ Other

Select a reason for disabling security defaults. Pay heed to the warnings here.

Instead of disabling security defaults across the organization you can apply a conditional access rule to the user account to enable basic authentication. This configuration is beyond the scope of this document. Ask your Azure Active Directory administrator about setting up conditional access rules.

The recommended method of authentication for the SMTP server is to use OAuth2 authentication in the same way as you configured authentication to the mail server to read mail.

This will require a slightly different TURNOVER authentication type to be configured and used when specifying the authentication type in the 'Error handling' section of the TURNOVER mailbox configuration.

The tinboxd/config folder in the location where you installed the TINBOXD software contains a sample properties file named OUTLOOK\_OAUTH\_SMTP.props that can be used to create this authentication type. Fill in the client\_id, client\_secret and tenant values that you used for the mail server authentication type.

Note that the SMTP port is defined as the secure port 587 but the protocol is set to plain SMTP. The authentication type must define the property 'mail.smtp.starttls.enable=true' in order to start a secure session with the server after connection has been established. Do not set the protocol to secure SMTPS in the mailbox error handling panel.

This concludes your configuration of OAuth authentication for the TURNOVER monitor application with Microsoft Entra ID. Continue reading this document from section 'Message Handling'.



---

## Google Cloud

The TINBOXD monitor connects to mail servers using IMAP or POP protocols. Google restrict the use of certain OAuth authentication flows when connecting to mailboxes using these protocols. As a consequence there are only two authentication methods available; authorization code flow and authentication using an application service account.

If you choose to use the authorization code flow when accessing a Google Mail server via IMAP or POP this will require a manual step to run TINBOXD from a Windows server initially to complete the authentication process. This method has the benefit of limiting the access to just a single mailbox and requires the express authorization of the mailbox user by insisting that they sign on with their password and accept the authorization request.

If you choose to use an application service account this simplifies the authentication process greatly since you no longer need to perform the manual authorization step. The service account can authenticate for any user. Because of this additional security measures are implemented such as limiting the access to a specific scope (the Google mail service) and requiring the TURNOVER mail monitor to have access to a secure public/private key to validate the use of the account.

For both of these authentication methods you will need to have your Google Cloud administrator create an application and assign to it the required permissions (scope).

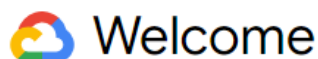
For authorization code flow a client identifier and client secret will be required to identify the application to the authentication server.

For service account authentication a public/private key pair is required packaged in a JSON file that must be made available to the TURNOVER mail monitor job at run time.

The first step is to create an application in the Google Cloud console.

## Create an application

Log in to your Google Cloud administrator console and work with APIs & Services:



You're working in [unicomglobal.com](#)

[Dashboard](#) [Recommendations](#)



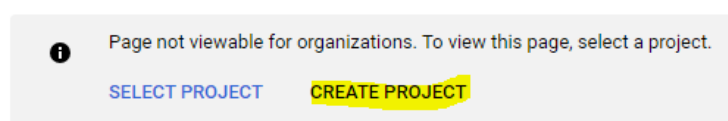
### Quick access



Work with projects and add a new project with the suggested name TURNOVER-TINBOXD:

### APIs & Services

#### Enabled APIs & services



## New Project

 You have 9 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[MANAGE QUOTAS](#)

Project name \*

TURNOVER-TINBOXD

?

Project ID: turnover-tinboxd. It cannot be changed later. [EDIT](#)

Organization \*

unicomglobal.com

▼

?

Select an organization to attach it to a project. This selection can't be changed later.

Location \*

 unicomglobal.com

[BROWSE](#)

Parent organization or folder

[CREATE](#) [CANCEL](#)

Select the new project as your working project and from the Google Cloud dashboard go to APIs & Services -> OAuth consent screen.

You can proceed in one of two ways:-

- Choose to make your app available to internal users only to prevent the need to have the app. verified by Google. This means you app will only be accessible to people in your organization.
- Or you can choose to make your app available to external users which will require you to go through a detailed verification process with Google but your app will be available to all users with a Google account.

To avoid the verification step you can set your app as external and set the publishing status to testing. You must then list the specific users that you wish to make the app available to. This is a good compromise if you just want to allow access to one or two mailboxes.

Choose which option you would like to use and click the Create button:

## OAuth consent screen

Choose how you want to configure and register your app, including your target users. You can only associate one app with your project.

### User Type

☒ Internal ?

Only available to users within your organization. You will not need to submit your app for verification. [Learn more about user type](#)

☐ External ?

Available to any test user with a Google Account. Your app will start in testing mode and will only be available to users you add to the list of test users. Once your app is ready to push to production, you may need to verify your app. [Learn more about user type](#)

CREATE

Enter suitable app. information. A suggested name is 'TURNOVER-TINBOXD'. You will need to enter suitable URLs to point to a home page, privacy and terms of service pages.

## Edit app registration

1 OAuth consent screen — 2 Scopes — 3 Summary

### App information

This shows in the consent screen, and helps end users know who you are and contact you

TURNOVER-TINBOXD

The name of the app asking for consent

User support email \*

██████@██████.com

For users to contact you with questions about their consent

### App logo

This is your logo. It helps people recognize your app and is displayed on the OAuth consent screen.

After you upload a logo, you will need to submit your app for verification unless the app is configured for internal use only or has a publishing status of "Testing". [Learn more](#)

BROWSE

Upload an image, not larger than 1MB on the consent screen that will help users recognize your app. Allowed image formats are JPG, PNG, and BMP. Logos should be square and 120px by 120px for the best results.

## App domain

To protect you and your users, Google only allows apps using OAuth to use Authorized Domains. The following information will be shown to your users on the consent screen.

Application home page

<https://softlanding.com>

Provide users a link to your home page

Application privacy policy link

<https://www.softlanding.com/softlanding-root/privacy/>

Provide users a link to your public privacy policy

Application terms of service link

<https://www.softlanding.com/softlanding-root/legal/>

Provide users a link to your public terms of service

## Authorized domains ?

When a domain is used on the consent screen or in an OAuth client's configuration, it must be pre-registered here. If your app needs to go through verification, please go to the [Google Search Console](#) to check if your domains are authorized. [Learn more](#) about the authorized domain limit.

Authorized domain 1 \*

[softlanding.com](https://softlanding.com)

[+ ADD DOMAIN](#)

The next step is to add the permissions or scopes that you want to grant to the application. To read email from a Google mail account using IMAP or POP protocols you will need to add the scope 'https://mail.google.com/'.

If the scope is not available in the list then you may need to enable the Google Mail API in the API Library. Click the Google API Library link to do this:

**i** Only scopes for enabled APIs are listed below. To add a missing scope to this screen, find and enable the API in the [Google API Library](#) or use the Pasted Scopes text box below. Refresh the page to see any new APIs you enable from the Library.

| Filter <span>API : Gmail API</span> <span>Enter property name or value</span> <span>X</span> <span>?</span> |           |                                                                 |                                                                       |
|-------------------------------------------------------------------------------------------------------------|-----------|-----------------------------------------------------------------|-----------------------------------------------------------------------|
|                                                                                                             | API       | Scope                                                           | User-facing description                                               |
| <input checked="" type="checkbox"/>                                                                         | Gmail API | <a href="https://mail.google.com/">https://mail.google.com/</a> | Read, compose, send, and permanently delete all your email from Gmail |

The new scope will be listed under restricted scopes:

## 🔒 Your restricted scopes

Restricted scopes are scopes that request access to highly sensitive user data.

### Gmail scopes

| API ↑ | Scope        | User-facing description                      |    |
|-------|--------------|----------------------------------------------|----|
| Gmail | https://mail | Read, compose, send, and                     |    |
| API   | .google.com/ | permanently delete all your email from Gmail | 🗑️ |

[SAVE AND CONTINUE](#)

[CANCEL](#)

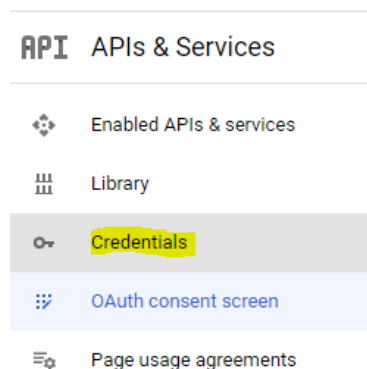
At this point, if you chose to set your application as externally available you will need to enter the email addresses of the users that you want to be able to access the app. on the next panel.

Finally you will be presented with a summary of your application registration. Return to the dashboard.

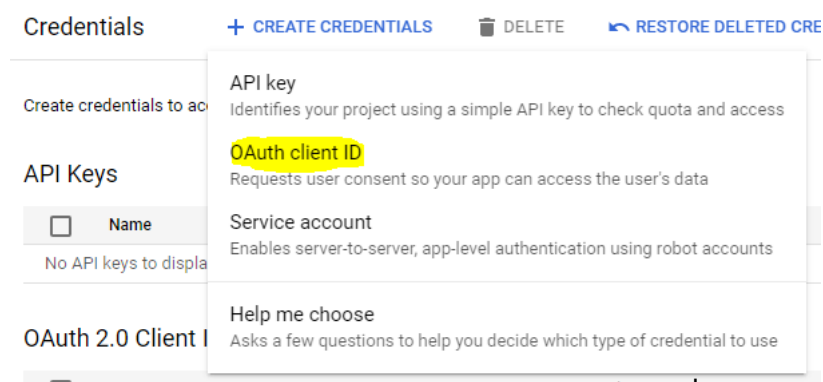
You now need to create suitable credentials that will be used as the client identifier and client secret or the service account public/private keys (PPK) JSON file in your OAuth authentication requests.

## Authorization code grant flow credentials

From the Google Cloud APIs & Services panel choose the 'Credentials' option:



Click on '+ Create Credentials' and choose OAuth client ID:



The Create OAuth client ID panel will be displayed.

## Create OAuth client ID

A client ID is used to identify a single app to Google's OAuth servers. If your app runs on multiple platforms, each will need its own client ID. See [Setting up OAuth 2.0](#) for more information. [Learn more](#) about OAuth client types.

Application type \*  
Web application ▼

Name \*  
TURNOVER-TINBOXD

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

 The domains of the URIs you add below will be automatically added to your [OAuth consent screen](#) as [authorized domains](#).

### Authorized JavaScript origins

For use with requests from a browser

 ADD URI

### Authorized redirect URIs

For use with requests from a web server

URIs 1 \*  
<http://localhost:8080/oauth>

 ADD URI

Note: It may take 5 minutes to a few hours for settings to take effect

Choose the Web application type and give the credentials a name such as 'TURNOVER-TINBOXD':

Add an authorized redirect URI. This URI should be for the localhost domain only. Whatever value you set here must be set in the 'redirect\_uri' property of the associated TURNOVER authentication type. We recommend that you set the redirect URI to 'http://localhost:8080/oauth'

Click the 'Create' button and the generated client ID and secret will be presented.

## OAuth client created

The client ID and secret can always be accessed from Credentials in APIs & Services

**i** OAuth access is restricted to users within your organization unless the [OAuth consent screen](#) is published and verified

|               |                                                                          |                                                                                     |
|---------------|--------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| Client ID     | 503389021393-gm3itgtvqm7ijbe3ualr0lventbbbvbm.apps.googleusercontent.com |  |
| Client secret | G7C3PX 3j-Lpjl.og.Vpnm."X^"-C 4k 3f+pC6.3                                |  |
| Creation date | October 5, 2023 at 2:51:22 PM GMT+1                                      |                                                                                     |
| Status        | ✓ Enabled                                                                |                                                                                     |

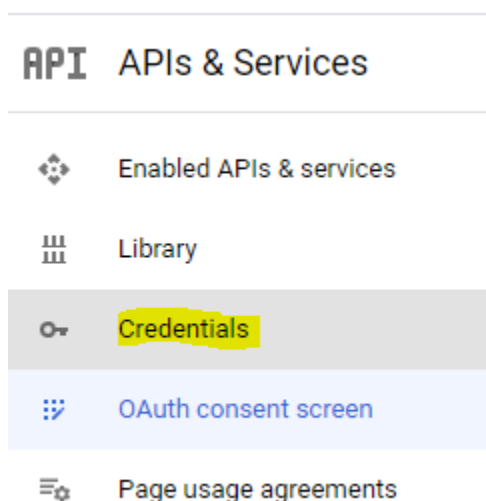
 [DOWNLOAD JSON](#)

Unlike Azure Entra ID you are able to save these credentials as a JSON file for future reference but we would discourage doing so as anyone who can access your client secret could make authentication requests through this app.

Copy the details and transfer them to your TURNOVER authentication type configuration to properties 'client\_id' and 'client\_secret' and immediately encrypt the client secret value there.

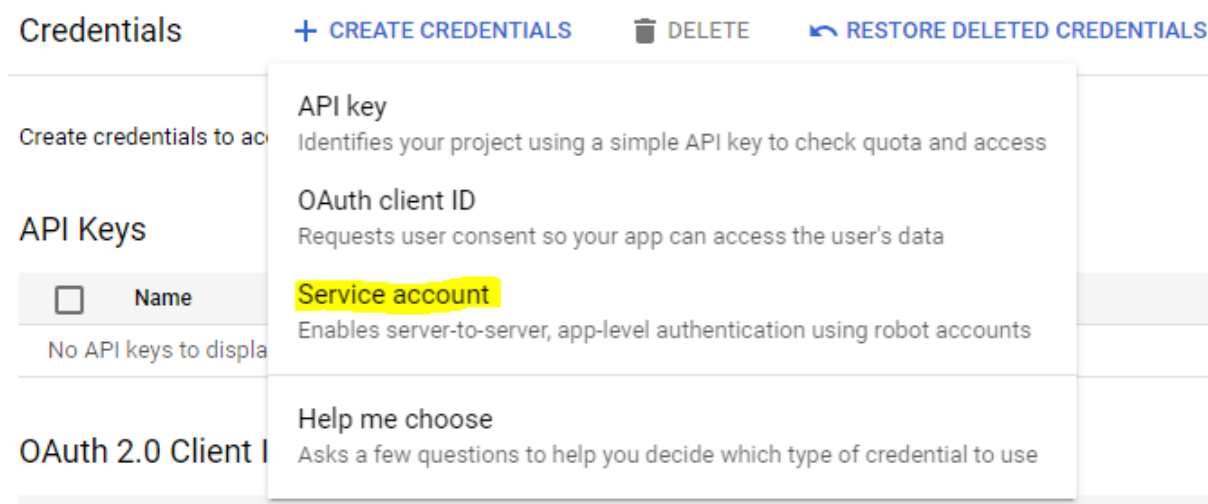
## Service account credentials

If you have chosen to authenticate using an application service account then from the Google Cloud APIs & Services panel choose the 'Credentials' option:



From the credentials panel click on '+ Create Credentials' and choose the service account option.





Enter suitable service account name, identifier and description. Your Google Cloud administrator may have required naming conventions for these otherwise you can choose any value.

[←](#) **Create service account**

parent page

**1 Service account details**

Service account name   
Display name for this service account

Service account ID \*  ✕ ↺  
Email address: turnover-tinboxd@turnover-tinboxd.iam.gserviceaccount.com 📧

Service account description   
Describe what this service account will do

**CREATE AND CONTINUE**

Click on create and continue then skip the next two optional sections of the service account registration.

Review the details of your new service account and note the email address and unique id that has been assigned to the account.

←
TURNOVER-TINBOXD

Previous page

PERMISSIONS

KEYS

METRICS

LOGS

### Service account details

Name

TURNOVER-TINBOXD

SAVE

Description

Service account for TURNOVER TINBOXD mail monitor

SAVE

Email

turnover-tinboxd@turnover-tinboxd.iam.gserviceaccount.com

Unique ID

102882889145318639168

Now create a public/private key (PPK) pair for this service account.

Return to the the APIs & Services -> Credentials panel and edit the service account:

Service Accounts [Manage service accounts](#)

| <input type="checkbox"/> | Email                                                                                                                                    | Name ↑           | Actions                                                                                                                                                                     |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <input type="checkbox"/> | <a href="mailto:turnover-tinboxd@turnover-tinboxd.iam.gserviceaccount.com">turnover-tinboxd@turnover-tinboxd.iam.gserviceaccount.com</a> | TURNOVER-TINBOXD |   |

From the service account details choose the Keys option:

[←](#) **TURNOVER-TINBOXD**

DETAILS

PERMISSIONS

KEYS

METRICS

LOGS

### Service account details

Name

TURNOVER-TINBOXD

SAVE

Description

Service account for TURNOVER TINBOXD mail monitor

SAVE

Email

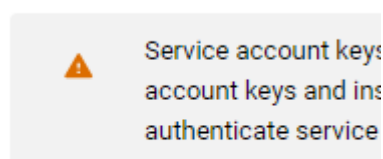
turnover-tinboxd@turnover-tinboxd.iam.gserviceaccount.com

Unique ID

102882889145318639168

Choose the ADD KEY option and Create new key:

## Keys



Add a new key pair or upload a p

Block service account key creati

[Learn more about setting organi](#)

ADD KEY ▾

Create new key

Upload existing key

Select the JSON option and click Create:

## Create private key for "TURNOVER-TINBOXD"

Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.

### Key type

☒ JSON

Recommended

☐ P12

For backward compatibility with code using the P12 format

CANCEL

CREATE

A PPK JSON file will be downloaded to your computer usually into the 'Downloads' folder but check your browser download location.

## Private key saved to your computer



pure-mission-401515-01ada72eb3f1.json allows access to your cloud resources, so store it securely. [Learn more best practices](#)

CLOSE

The file will have a name in the form 'xxxx-xxxxxxx-401515-01ada72eb3f1.json'. The letters and numbers may be different but the format comprising two words, a number and hex string with the .json suffix will be the same. If you open the file with a text editor the contents will be similar to this (note: the private key in this example has been cut down in length):

```
{
  "type": "service_account",
  "project_id": "xxxx-xxxxxxx-401515",
  "private_key_id": "01ada72eb3f17d26f49a9f058988a727516dfdbc",
  "private_key": "-----BEGIN PRIVATE KEY-----
\nMIIEvAIBADANBgkqhkiG9w0BAQEFAASCBywggSiAgEAAoIBAQCmHStAdld69iNU\nnjyJ3kED
u7eDPqPqmgxRy0+hN1z04H6+AR5\nGMvAp80kxMDgDw1+kZtvfg==\n-----END PRIVATE
KEY-----\n",
  "client_email": "turnover-tinboxd@xxxx-xxxxxxx-
401515.iam.gserviceaccount.com",
  "client_id": "100217445323306432821",
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  "token_uri": "https://oauth2.googleapis.com/token",
  "auth_provider_x509_cert_url":
"https://www.googleapis.com/oauth2/v1/certs",
  "client_x509_cert_url":
```

```
"https://www.googleapis.com/robot/v1/metadata/x509/turnover-tinboxd%40xxxx-xxxxxxx-401515.iam.gserviceaccount.com",  
  "universe_domain": "googleapis.com"  
}
```

You must keep this file secure. If it should ever be compromised you can create a new key file to replace it. You should transfer this file to a secure location in the IFS on your IBM i server or the server where you intend to run the TINBOXD monitor process. The file should have restricted public access and only given read permissions to the user that runs the TINBOXD monitor job.

The final step is to allow this service account to access your Google Workspace domain. This must be carried out in your Workspace administration console. You will need the service account unique ID that can be found in the advanced settings of the service account details here:

## Advanced settings



### Domain-wide Delegation



Granting this service account access to your organization's data via domain-wide delegation should be used with caution. It can be reversed by disabling or deleting the service account or by removing access through the Google Workspace admin console.

[LEARN MORE ABOUT DOMAIN-WIDE DELEGATION](#)

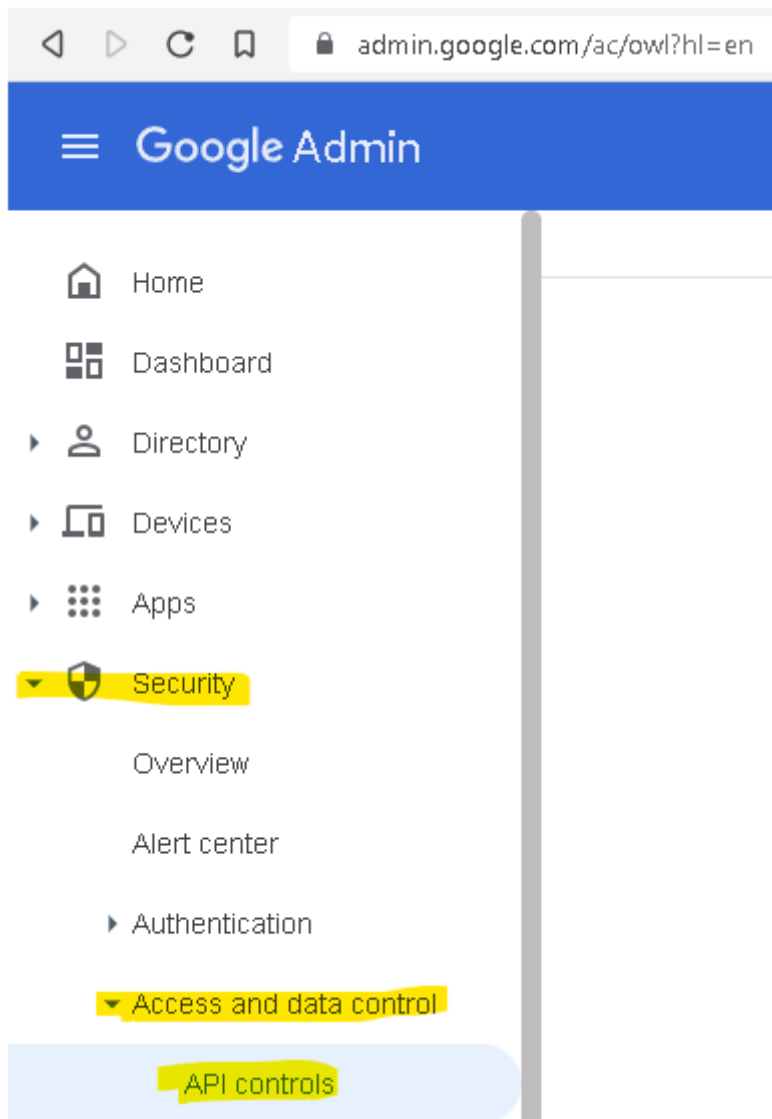
Client ID: 102882889145318639168



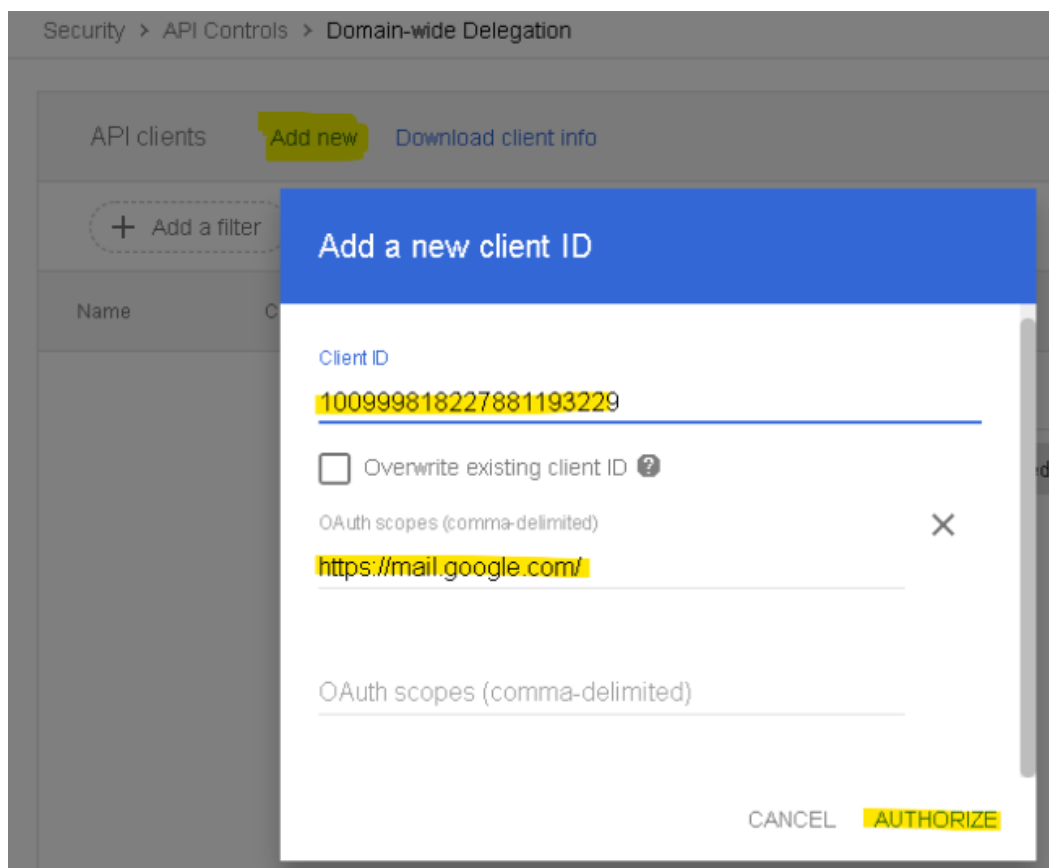
[VIEW GOOGLE WORKSPACE ADMIN CONSOLE](#)

You can also go directly to the Google Workspace admin console from the button presented here.

From the navigation panel go to Security -> Access and data control -> API controls -> Manage domain wide delegation:



Add a new entry using the client id and OAuth scopes required which should be either <https://mail.google.com> or [https://www.googleapis.com/auth/gmail.imap\\_admin](https://www.googleapis.com/auth/gmail.imap_admin).



This concludes the authentication configuration process.

## Initial Google Mail account authentication

If you have chosen to create an application service account and granted it permissions and domain wide access to mail then the following step is not necessary. Continue reading from section 'Google mail POP/IMAP configuration'.

If you have chosen to create an application and you have created client credentials (client id and client secret) you need to use the OAuth authorization code flow to initially authenticate access to Google Mail using IMAP or POP protocols. To achieve this it will be necessary to run the TINBOXD client from a Windows server to establish the initial authentication and acquire access and refresh tokens. The reason for this is that the authorization code flow requires access to a graphic user interface and web browser. These are not available in the native IBM i server environment.

Once this step has been performed it will not need to be performed again until the refresh token expires (possibly for up to 3 years) or the mailbox user changes their password or revokes the authorization.

Use the TURNOVER standalone client or the RDi perspective to create the necessary authentication type and apply this to a mailbox definition as described earlier in this document.

In order to run the TINBOXD process from a Windows server you will need to have Java installed on the Windows server and map a shared folder on the IBM i server to the IFS folder where the TINBOXD software is installed onto a Windows drive letter. This is the IBM i folder '/turnover/tinboxd' by default.

On the IBM i the Netserver TCP server needs to be started to enable mapped drives to be established. The Netserver can be started by issuing the following command from a 5250 session or from the Operations Navigator console.

```
STRTCPSVR SERVER(*NETSVR)
```

On the Windows server open a command or Powershell terminal and map the network drive as follows:

```
net use X: \\<ibmi_host>\<share>\turnover\tinboxd
```

Where X: is an available drive letter (run command 'net use' to see which drive letters are unused), <ibmi\_host> is your IBM i server host name, <share> is the name given to the shared folder by the Netserver configuration and 'turnover\tinboxd' is the location of the TINBOXD software if your share name does not already refer to this folder directly.

Switch to that mapped drive letter and list the folder content to check that you are in the software install folder. You should see the file tinboxd.jar in the listed files.

Because you are running the TINBOXD process on a different server than the IBM i host server you will need to have suitable remote IBM i host server name, user and password values stored in a properties file before you run the TINBOXD monitor. If you have installed TURNOVER in alternative library names you will also need to set the appropriate command, data and program library names and a suitable TURNOVER environment.

See the previous sections in this document regarding the required properties and make sure you have a properties file that has values for AS400System, AS400User and AS400Password that are NOT 'localhost' and '\*CURRENT' in the current folder.

To invoke the TINBOXD process to authenticate mailbox users run the following command:

```
java -DauthenticateOnly -DclearTokenCache -jar tinboxd.jar  
ibmi.properties
```

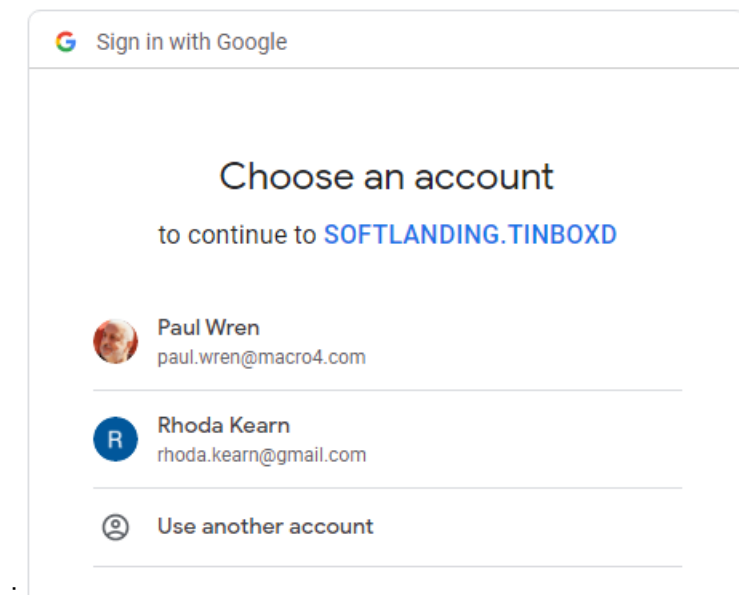
The authenticateOnly property will tell the monitor process to just authenticate the active mailboxes and not actually read and process the mailbox contents. The clearTokenCache property will wipe out any currently cached access tokens and force authentication to be performed again for all active mailboxes.

The file ibmi.properties contains the required property values for the IBM i host, user and password and any non-standard TURNOVER library names.

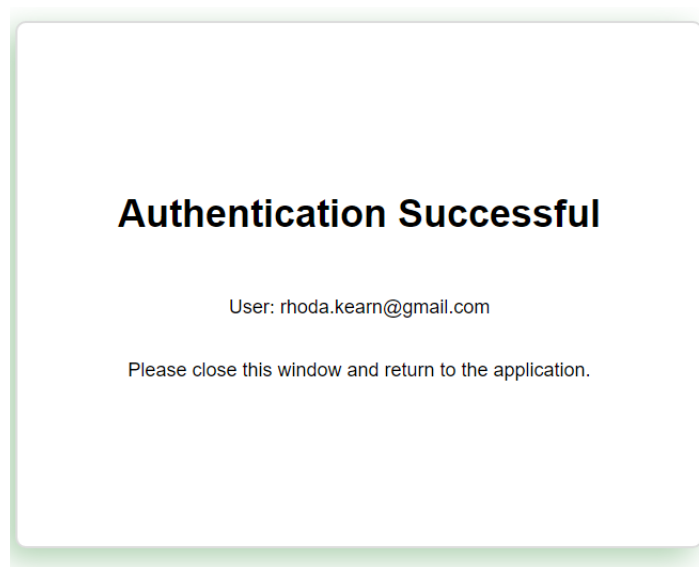
For each active mailbox configured in TURNOVER the TINBOXD process will attempt to authenticate the user's access to the server and mailbox.



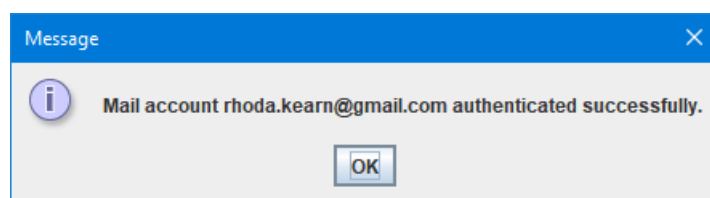
For the Google mail authentication or any other authorization code flow a web browser will be automatically opened showing the necessary confirmation panels to complete the authentication. For Google Mail the following page will appear if you have multiple Google accounts:



Choose the required account that you want to authenticate the application. If the authentication request is successful then the following page will be displayed in the web browser:



A pop up dialog stating that the authentication was either successful or unsuccessful will be displayed.



If the authentication was successful then the required access and refresh tokens will have been stored in the tokens.json cache file and will be used to authenticate access to the mailbox on subsequent connection attempts by the TINBOXD processing.

You can now restart the TINBOXD process on the IBM i server as normal to proceed with mail inbox monitoring.

If there are any problems authenticating the user then you will need to analyze the log output file to establish the cause of the problem. You may also need to check the 'Enable mail debug logging' box on the mailbox configuration to increase the level of logging output and then run the authentication process again.

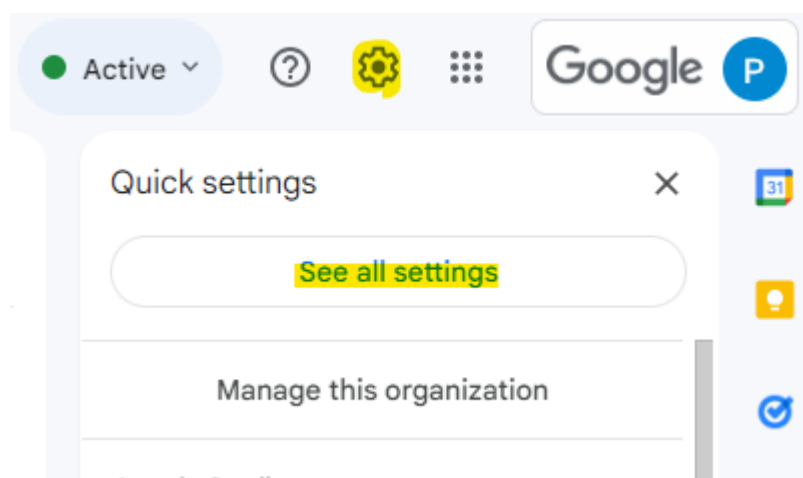
This should conclude your configuration of OAuth authentication for the TURNOVER monitor application with Google Mail. Continue reading this document from section 'Message Handling'.

---

## Google mail POP/IMAP configuration

In order to be able to access a Google Workspace user's mailbox using the IMAP or POP protocols you must enable these protocols in the user's mail settings as follows:

Log in as the required user and open Gmail. Click the settings gear icon and choose to 'See all settings':



The settings panel will be displayed:

Accounts   Filters and Blocked Addresses   **Forwarding and POP/IMAP**   Add-ons   Chat and Meet

---

Tip: You can also forward only some of your mail by [creating a filter!](#)

---

**1. Status: POP is disabled**  
☒ Enable POP for all mail  
☐ Enable POP for mail that arrives from now on

**2. When messages are accessed with POP**

**3. Configure your email client** (e.g. Outlook, Eudora, Netscape Mail)  
[Configuration instructions](#)

---

**Status: IMAP is enabled**  
☒ Enable IMAP  
☐ Disable IMAP

**When I mark a message in IMAP as deleted:**  
☐ Auto-Expunge on - Immediately update the server. (default)  
☒ Auto-Expunge off - Wait for the client to update the server.

**When a message is marked as deleted and expunged from the last visible IMAP folder:**  
☐ Archive the message (default)  
☒ Move the message to the Trash  
☐ Immediately delete the message forever

**Folder size limits**  
☒ Do not limit the number of messages in an IMAP folder (default)  
☐ Limit IMAP folders to contain no more than this many messages

Select the 'Forwarding and POP/IMAP' settings.

Select 'Enable IMAP' and/or 'Enable POP for all mail' options.

If you are receiving messages using the IMAP protocol from the top level INBOX folder then you can set Auto-Expunge on to handle deleted messages. You can then choose how you want the server to handle the deleted messages.

If you are using your own named folder or nested sub folders to be sure that messages accessed via IMAP that are successfully handled by the TURNOVER mail monitor are deleted correctly from the server, turn off the Auto-Expunge option. The TURNOVER mail monitor will use the authentication type properties 'empty\_trash' and 'trash\_folder' to determine how to handle deleted messages. If you want the monitor to move handled messages to the trash folder use the property 'trash\_folder' to name the folder where the messages are to be moved to. For Google mail this is usually '[gmail]/trash' but may be different.

If you do not specify the name of a trash folder and you are monitoring a named folder or nested sub folder then messages will be marked as deleted but may still be retained on the server. This is a feature of the Google mail management of IMAP folders and labels.

To have the monitor job empty the named trash folder after each poll set the 'empty\_trash' property to true. The default is false.

---

## Message handling

Before handling messages, TURNOVER Mail Integration logs into each of the configured mailbox accounts and checks the specified mailbox folder for new mail. If new mail exists, the messages are collected and the handling process begins for each one in turn. First, a TURNOVER task Id is searched for in the message headers. If a task Id is not found, the message follows the sequence for adding a new task. If a task Id is found, the message follows a sequence for appending to the task details.

---

### Finding an email and task association

Determining a correlation between the email and an existing TURNOVER task is the start of message handling. When TURNOVER generates an email message, a custom message Id consisting of the original message Id, the TURNOVER task Id, and the constant string "@turnover.helpdesk" is created. This custom message Id is substituted for the original message Id. Through normal email handling by the email server, this new message Id is added as a value to one or both of the email headers "In-Reply-To" and "References" as the message is sent out. The task Id is also referenced in the email subject line. When the message recipient replies to the email, the custom message Id is automatically passed back to TURNOVER in the email headers, once again through normal handling by the email server.

When the email message is examined for an existing task Id, the email headers are first searched for the "In-Reply-To" header. If the header contains a value, the value is compared to the constant string "@turnover.helpdesk". If this string is found within the header value, then the task Id is pulled from the remainder of the value text. The task Id is then compared to the text in the subject line. If the subject contains a match to the task Id, an association has been established.

If the "In-Reply-To" header does not exist or contain a value, then the header "References" is searched for. If this header contains a value, the same steps above are taken. The value is compared to the constant string, and if found, the task Id is extracted from the remaining text. If the same task Id is also found in the subject line, an association has been established.

In some cases, the email server may not set the headers at all. To combat this, TINBOXD allows the configuration of a regular expression to search the email subject line and extract the task Id.

Confirming a task association is still not a guarantee that the email is related to an existing TURNOVER task. The task Id must be checked against the database to confirm that the task Id belongs to a valid task. If there is not a valid TURNOVER task for the task Id, this is an error condition and the message is forwarded for manual handling. An invalid task result contrasts with an unconfirmed task association. If the headers do not contain appropriate values, or if the subject line does not contain a task Id match, then a task association is not established. In this case, a new task is added for the email message.

---

## Replying to a TURNOVER message

It is important to note that the message generated by TURNOVER regarding an existing task must be directly replied to, using the email client's routine "reply to" options, in order for TURNOVER Mail Integration to work properly. If, for example, the email is forwarded to another address, and then emailed back to the TURNOVER address, this will break the process chain. TINBOXD depends on standard email server practices to pass the proper email headers and their values. The only way the proper header values will return to TINBOXD is if the message comes back via a direct email "reply".

Most email clients provide the option of replying to a message with or without including the text of the original message, or the message history as it is typically called. Both reply methods will result in TINBOXD adding the message text to the task details. However, if the reply contains the message history, the history will be included in the task details. Since the original message is generated using the task details, this will produce duplication in the details. It is better to reply to the TURNOVER generated message without the message history.

---

## Adding a new task

In preparation for adding a new task, the task defaults for the project specified in the `defaultProject` property are retrieved. Next, user authorization is checked. If the `turnoverUser` property is not `*CURRENT`, the user provided must have add task authority to the project. If `*CURRENT` is specified, the user provided for the `as400User` property must have add task authority to the project. If the user is authorized, the task creation process begins:

- The message subject is retrieved and becomes the task description.
- The requester is found by using the email address of the message sender. If the email address matches a requester email address, the requester Id is used in the task.
- A task details comment header is constructed from a current time stamp and the requester Id.
- The task details begin with header block containing the date the email was sent, the email sender's address and the email subject. The message text is parsed for plain text, and the resulting text becomes the task details.

- Attachments included with the email message are checked against the list of excluded extensions in the `ignoreAttachmentFileExtensions` property. If the attachment extension is not included in the list, the attachment is added to the task as a linked document.
- The task defaults from the project fills in the remaining task fields. If the task default for the resource is `*USRPRF`, the resource is the profile determined previously by the `turnoverUser` and `as400User` property values.

---

## Appending details to an existing task

After finding a task Id in an email header, the email subject is also checked for the presence of the task Id. Alternatively, the supplied regular expression is used for extracting the task Id from the subject. If the conditions are satisfied, verification of the existence of the task is performed. If a TURNOVER task is found for the task Id, the change task process begins:

- The task is first checked for being in use by another person. If the task is locked, the email is not handled. It remains on the server and is collected by TINBOXD again the next time around. This iteration continues until the task is unlocked and the message can be processed.
- User authorization is checked in a similar manner as previously mentioned for adding a new task. If the `turnoverUser` property is not `*CURRENT`, the user provided must have change task authority to the project. If `*CURRENT` is specified, the user provided in the `as400User` property must have change task authority to the project. If the user is authorized, the change task process continues.
- The task is next checked for status. If the task is closed, the email is forwarded for manual handling. If the task is open, the message continues processing.
- A task details comment header is constructed from a current time stamp and the email address of the message sender. If the sender is an existing requester, the requester Id is displayed. If the sender is not a requester, part of the sender's email address is displayed.
- The appended detail text begins with a header block containing the date the email was sent, the email sender's address and the email subject. The message text is parsed for plain text, and the resulting text becomes the task details.
- Attachments included with the email message are checked against the list of excluded extensions in the `ignoreAttachmentFileExtensions` property. If the attachment extension is not included in the list, the attachment is added to the task as a linked document.
- The append details exit program is called if the `appendExitPgm` and `appendExitLib` properties specify an exit program.

---

## Message text parsing

Various components of each email message are placed in various elements of a task. For example, as previously mentioned, the email subject text becomes the task description. The body of the message is separated into task details and possibly links, depending on the structure of the received email. An ordinary text component is saved as task details. Attachment components are converted to task links. In-line components, such as embedded images or other special blocks of content, are also converted to task links. There is a difference between the attachment links and the in-line links. The attachment components contain a file name, and that file name is assigned to the task link. The in-line components contain only a type definition of the content, without a file name. In these cases, a stock file name for the task link is generated using the content type.

---

## Message forwarding

Email messages are not automatically processed for cases such as the task associated with the email is closed, or when adding a new task and a requester equating to the email sender is not found, or when an error occurs while processing the message information. Under these circumstances, the emails are forwarded for manual handling. As noted previously, in cases when the task is open for editing, thus preventing the message text from appending to the task details, the email is not forwarded. The message remains on the server and automatic handling by TINBOXD continues with each pass until the task is unlocked and the message can be processed.

Messages need not be forwarded to the same email address. There are four mailbox properties that can be used for directing email forwarding, depending on the situation. The `smtp.closedTaskForward` property specifies an address for forwarding messages when the task associated with the email is closed. The `smtp.requesterForward` property specifies an address for forwarding messages when a requester cannot be found for the email sender. The `smtp.errorsForward` property specifies an address for forwarding messages in case of general processing errors. Some of these errors can include: the connection user is not authorized to perform the necessary TURNOVER task functions; the message text is in a format, such as HTML, that cannot be converted to plain text for the task details; an unexpected error occurs during any point in the task creation/update process. The `smtp.defaultForward` property provides an optional address for any, or all, of the other three address properties. A blank value for any, or all, of them will default the forward to the `smtp.defaultForward` property address.

The `smtp.defaultForward` property also acts as a default address if any of the other three address properties cannot be resolved. Having such responsibility, the `smtp.defaultForward` property is the only one of the four that absolutely must be a single valid email address. The other three address properties can be either a single valid email address, or one of the special field values as listed previously under the property descriptions.

Dividing the forwarded messages to various addresses gives TINBOXD the flexibility to blend with an established work flow. For example, messages coming from an email sender who does not exist as a requester might be forwarded to the sales team for adding the sender into a company system; or forwarded to the project coordinator for adding the sender into TURNOVER.

Closed task messages might be forwarded to the task resource to give notice of continued interest in the task; or for internal communications, forwarded to the email sender to alert them of the task status. Messages due to errors might be forwarded to a TURNOVER administrator, or systems administrator. For the ease of administration, all three address properties can be set to the same address, or better yet to the same special value, such as the coordinator. In this case, the person set as the coordinator in TURNOVER can be easily changed during vacations or other absences, and no change needs to be made to the setup of TINBOXD.

---

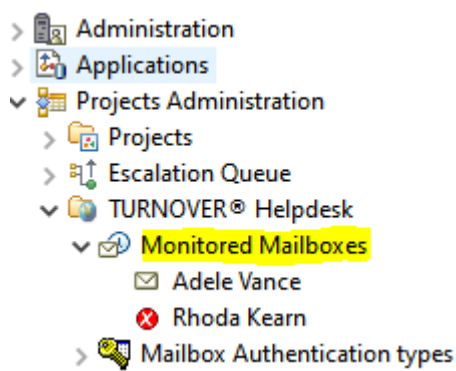
## Setting up TURNOVER mail integration

The first step in setting up TINBOXD is to decide for which TURNOVER projects email communications would be good candidates for automatic handling by TINBOXD. For instance, a Helpdesk project for customer requests, or a Helpdesk project for IT operations.

Next, identify an email address associated with incoming requests for the project and continued contact for its existing tasks. Ideally this email address should be dedicated to receiving mail for the designated project tasks. If multiple address are related to the project requests, a set of mailbox properties can be created to monitor each of the addresses. If an appropriate email address does not exist, an email account must be created.

For each of the identified email addresses, create a mailbox definition in the standalone client or RDi plugins or create a set of mailbox properties in the TINBOXD properties file, following the conventions in the configuration properties example and the properties descriptions as explained above.

We recommend that you define the mailboxes to be monitored using the standalone client or RDi plugins here:



If you are unable to configure the mailbox there, configuring TINBOXD with a properties file is not difficult. It is important that the property names and values are correct. Specifically, the mailbox property names must properly contain the numeric prefix notation depicted previously under mailbox configuration.

Each mailbox that is defined must be associated with an authentication type that determines how the user is authenticated to the mail server. The simplest authentication type is basic user/password authentication and that is represented by the 'PASSWORD' authentication type. For more secure authentication types such as OAuth2, other authentication types are provided but must be configured as described earlier in this document.



Once the mailboxes and connection properties are set, and the property file changes are saved, TINBOXD is ready to begin monitoring the specified email addresses. TINBOXD checks for new mail as often as is set in the `checkMailIntervalSeconds` property. Any changes made to the mailbox properties will be picked up by TINBOXD the next time it checks for new messages. The email monitor does not have to be stopped and restarted for the property modifications to take effect.

## Running the TINBOXD mail monitor

The TINBOXD monitor is a Java application that can be run on either a host IBM i or any other platform that supports a Java virtual machine and that can access the host IBM i via TCP/IP.

The normal configuration steps are to create a suitable job description to run the TINBOXD program and then add that job description to a suitable subsystem description so that it will automatically be started.

Alternatively the job can be scheduled using the job description and the IBM job scheduling facilities.

## TCFGTMI command

The command TCFGTMI (Configure TURNOVER mail inbox monitor) will perform some of these steps for you automatically. You simply enter the name and library of a job description you want to be created or used and the name and library of a suitable subsystem description and the command will add an auto-start job entry to that subsystem. The subsystem must exist and not already have an auto-start job entry using the job description name.

The RQSDTA of the job description that is created by the TCFGTMI command is set to 'TINBOXD' when a job description is created. This is sufficient to run the TINBOXD process on the local host with the default TURNOVER installation libraries. You may need to amend this value if you need to apply specific non default parameters to the TINBOXD command as described below.

| Parameter | Description                                                                                                                                                                                                                                                                                                                       |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FUNCTION  | <p><b>*CHECK</b></p> <p>Check that running the configuration will be successful. If any problem would prevent the configuration from completing, then messages will be displayed indicating the errors encountered. Correct the errors and try the command again.</p> <p><b>*RUN</b></p> <p>Perform the configuration proper.</p> |

| Parameter | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| USER      | <p>The user profile that the autostart job will run under.</p> <p>This user name is set in the job description definition as the USER parameter.</p> <p>If the user does not exist and the job description is to be created automatically then the configuration will fail.</p> <p><b>*CURRENT</b></p> <p>The user running the current job is used by default.</p> <p><b>User</b></p> <p>The name of the user profile to set in the job description.<br/>The user must exist.</p>                                                                                                                                                                                                                                                                                          |
| JOB       | <p>The name of the job description to be used or created to add the auto-start job entry to the named subsystem.</p> <p>If the job description already exists it will be used unchanged.</p> <p>If the job description does not exist one will be created with USER set to the USER parameter and the initial library list will be based on the current TURNOVER libraries found in the library list.</p> <p>If libraries cannot be determined from the current library list the defaults of SOFTTURN, SOFTTURND and SOFTTURNE will be used.</p> <p><b>*CURLIB</b></p> <p>The job description library. The default *CURLIB will use the current library set for this job.</p> <p><b>Name</b></p> <p>The default job description name TINBOXD or a name of your choice.</p> |
| SBSD      | <p>The name of the subsystem to add the autostart job entry to. The subsystem must exist.</p> <p><b>*CURLIB</b></p> <p>The subsystem library. The default *CURLIB will use the current library set for this job.</p> <p><b>Name</b></p> <p>The default subsystem name TINBOXD or a name of your choice.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

## TINBOXD command

The TINBOXD command is responsible for invoking the Java monitor process. It should usually be submitted as a batch process as it runs continuously checking the content of configured mailboxes at the desired interval.

The underlying Java process can also be run from a QSH or PASE terminal session using the Java – jar command option.

The TINBOXD command has been enhanced to provide some additional parameters to control the environment that will be set when invoking the Java program.

| Parameter | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PROPS     | <p>Identifies the location of a properties file in the IFS that describes the IBM i and TURNOVER® Lifecycle Manager connection attributes and optionally a set of properties for each mailbox that is to be monitored.</p> <p>If this file is in the same folder as that specified for PATH then the file can be referenced relative to that folder without the need to include the full path.</p> <p>The help text of the command contains a description of all of the possible properties that can be used in the file.</p> <p><b>*NONE</b></p> <p>If the monitor should run with all local properties for server and current user and mailbox definitions are maintained in the TURNOVER® Lifecycle Manager client then there is no need to provide a properties file.</p> <p><b>Name</b></p> <p>The optional path and properties file name.</p> |

| Parameter | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| LOGFILE   | <p>Specifies the path and file name of the log file that will be used to output stdout and stderr output streams from the Java process.</p> <p>Additional logging can be generated by amending the log4j2.xml configuration file stored in folder ./config in the folder where TINBOXD is installed (see PATH below).</p> <p>This requires a certain degree of knowledge of how to configure log4j and it is recommended you only amend this file under instruction from a support team member to acquire additional logging to provide for a support issue.</p> <p>If this file is in the same folder as that specified for PATH then the file can be referenced relative to that folder without the need to include the full path.</p> <p><b>*PRINT</b></p> <p>All output will be sent to a spooled print file.</p> <p><b>IFS path and file</b></p> <p>The optional path and file name in the IFS to receive log data. For example: logs/tinboxd.stdout.log</p> |
| PATH      | <p>The folder in the IFS where the TINBOXD software is installed.</p> <p><b>*DFT</b></p> <p>Use the default installation path /turnover/tinboxd.</p> <p><b>Name</b></p> <p>The full IFS path name where the TINBOXD software is installed.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

| Parameter | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ENV       | <p>Specify the name of the TURNOVER® Lifecycle Manager product library and the environment to be used to set an appropriate library list for the process.</p> <p><b>*JOB</b></p> <p>The product library and environment will be derived from the job description library list.</p> <p>Use this value when the job is being submitted to batch and the job description has already set the required libraries to be added to the job library list.</p> <p><b>Name</b></p> <p>The name of the library where TURNOVER® is installed and the environment to be used. By default this will usually be SOFTTURN/ENGLISH.</p> <p>Use this value when you are running the command interactively to authenticate mailboxes and you need to specify the installation of TURNOVER® that should be used or when you want to dynamically set the installation library and environment at runtime.</p> |
| CLRTOKEN  | <p>Determines whether the OAuth access token cache identified in the PROPS property file is to be cleared before starting the monitor process.</p> <p><b>Important</b> Clearing the token cache will mean that any existing access tokens that were previously acquired must be reacquired. This may mean responding to messages on the system operators message queue that may direct you to perform actions to complete the authentication process.</p> <p>Use the option AUTHONLY(*YES) to run TINBOXD in an interactive job or from a Windows PC to complete this process depending on the authentication types assigned to the mailboxes.</p> <p><b>*NO</b></p> <p>The OAuth access token cache should not be cleared.</p> <p><b>*YES</b></p> <p>The OAuth access token cache will be cleared.</p>                                                                                  |

| Parameter | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AUTHONLY  | <p>Determines whether the monitor process is only to perform the authentication steps for the specified mailboxes and not start the monitor process.</p> <p>Each mailbox that is defined will be contacted and authenticated to make sure that it can be logged in. The process will then end and return to the calling job.</p> <p>This is the only circumstance where the TINBOXD command can be invoked from an interactive command line. All other times the command must be submitted to batch to run as a background process.</p> <p>Look for messages on the system operator message queue that may require attention if the process takes some time to complete.</p> <p><b>*NO</b></p> <p>The process will start to monitor mailbox content and must be submitted as a batch job.</p> <p><b>*YES</b></p> <p>The process will attempt to authenticate each mailbox and then end.</p> |

## Running TINBOXD as a Java application

The TINBOXD process can be run as a plain Java application on any platform that supports a Java virtual Machine at version 1.8 (Java 8) or higher and has a TCP/IP connection with the IBM i server where TURNOVER is installed.

The process can be run on a Windows server by either mapping a shared folder using Windows net TCP server to share the TINBOXD installation folder on the host IBM i server or by copying the software from the IBM i installation folder to a suitable folder on the host server.

The process can also be run from a QSH or PASE (CALL QP2TERM) terminal session on an IBM server or with secure shell (ssh) from a remote terminal.

When running on a Windows server we recommend that you use the network server feature and create a mapped drive over the installation folder which is by default '/turnover/tinboxd'.

It may be necessary to run the application from a Windows server in order to complete any initial authentication process that may be required by your email service provider where more secure modern authentication methods are in place such as OAuth2.

We normally recommend that the TINBOXD process is run on the host IBM i server where TURNOVER is installed and only run it on a Windows server when authentication that requires access to desktop functions such as a web browser or dialog boxes are required such as the OAuth2 authorization flow.

The content of the installation folder on the IBM i after installing TURNOVER is as follows:

```
/turnover
+-> tinboxd
    +-> config
        | +-> GMAIL_OAUTH2_AUTH.props
        | +-> log4j2.xml
        | +-> OUTLOOK_OAUTH2_AUTH.props
        | +-> OUTLOOK_OAUTH2_CREDS.props
        | +-> OUTLOOK_OAUTH2_DEV.props
        | +-> OUTLOOK_OAUTH2_ROCP.props
        | +-> PASSWORD.props
        | +-> sample.properties
    +-> docs
        | +-> copyrights.txt
        | +-> TurnOver Mail Integration Specification.doc
    +-> lib
        | +-> (dependent library jars)
    +-> logs
        | +-> (default log file folder)
    +-> tinboxd.jar
    +-> sample.properties
```

The 'config' folder holds default property sets for the various supported authentication types that can be imported to create your own authentication types in the TURNOVER client.

The 'log4j2.xml' file contains logging configuration. If you are familiar with log4j2 as a logging service you may want to amend this file to set enhanced logging levels or change the location where the log files will be saved. The default is a folder named 'logs'.

The 'sample.properties' file is a sample file of the configuration properties that you may need to provide to the monitor process to control the IBM i server connection and the TURNOVER installation libraries etc. The properties are described in detail elsewhere in this document and also in the sample file and the TINBOXD IBM i command help text.

If you are running the monitor on a platform other than the same IBM i server that TURNOVER is installed on, or with different TURNOVER installation libraries from the standard default library names, you will need prepare a configuration properties file and specify it in the call to the Java process. Copy this sample file and edit it to add the values you need and then use the name of the copied file in the call to the Java process.

The 'docs' folder contains a copy of this document in MS Word format and an obligatory text file containing copyright notices of open source software used by the application.

The 'lib' folder contains the various Java libraries that the application is dependent upon.

The 'tinboxd.jar' file is the mail monitor application.

To run the monitor from a command line in Windows, open a command window and change the current directory to the mapped drive letter or the folder where you have copied the installation folder to, and run the following command:

```
Java -DauthenticateOnly -DclearTokenCache -jar tinboxd.jar  
configuration.properties
```

The `-DauthenticateOnly` property indicates that you want the monitor to check authentication is valid for each configured mailbox. With this property set and depending on the authentication type set in each mailbox configuration additional actions may be required to complete the authentication step. A web browser may automatically be opened, for example, to confirm user authentication to the TINBOXD application or messages may be sent to the host IBM i server QSYSOPR message queue that may need attention.

Once each mailbox has been authenticated the process will end returning you to the command prompt.

Omit this property if you want the monitor to process the configured mailboxes continuously. This will lock the terminal session until the process is cancelled.

The `-DclearTokenCache` property instructs the monitor process to delete all currently stored access tokens used for authentication and to re-authenticate each mailbox.

You may need to do this if you change the password of configured mailboxes or previously acquired refresh tokens have expired.

Omit this property if you do not want to clear the currently cached access tokens. If you retain access tokens that have expired or for users that have changed their mailbox passwords then authentication attempts may fail.

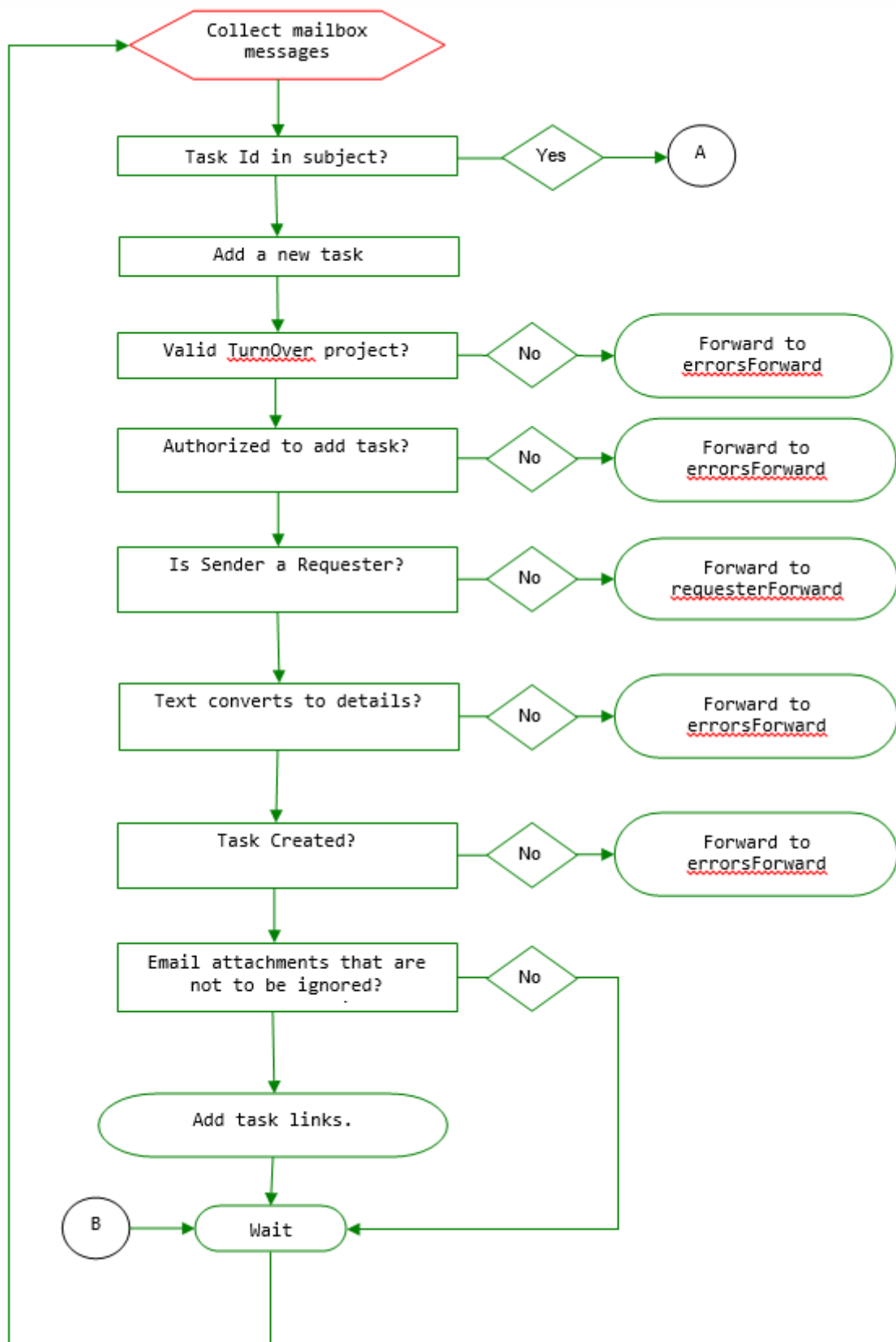
Cached tokens are stored in the folder that is indicated by the configuration property `tokenCacheFilename` set in the configuration properties file. The default filename is `tokens.json`.

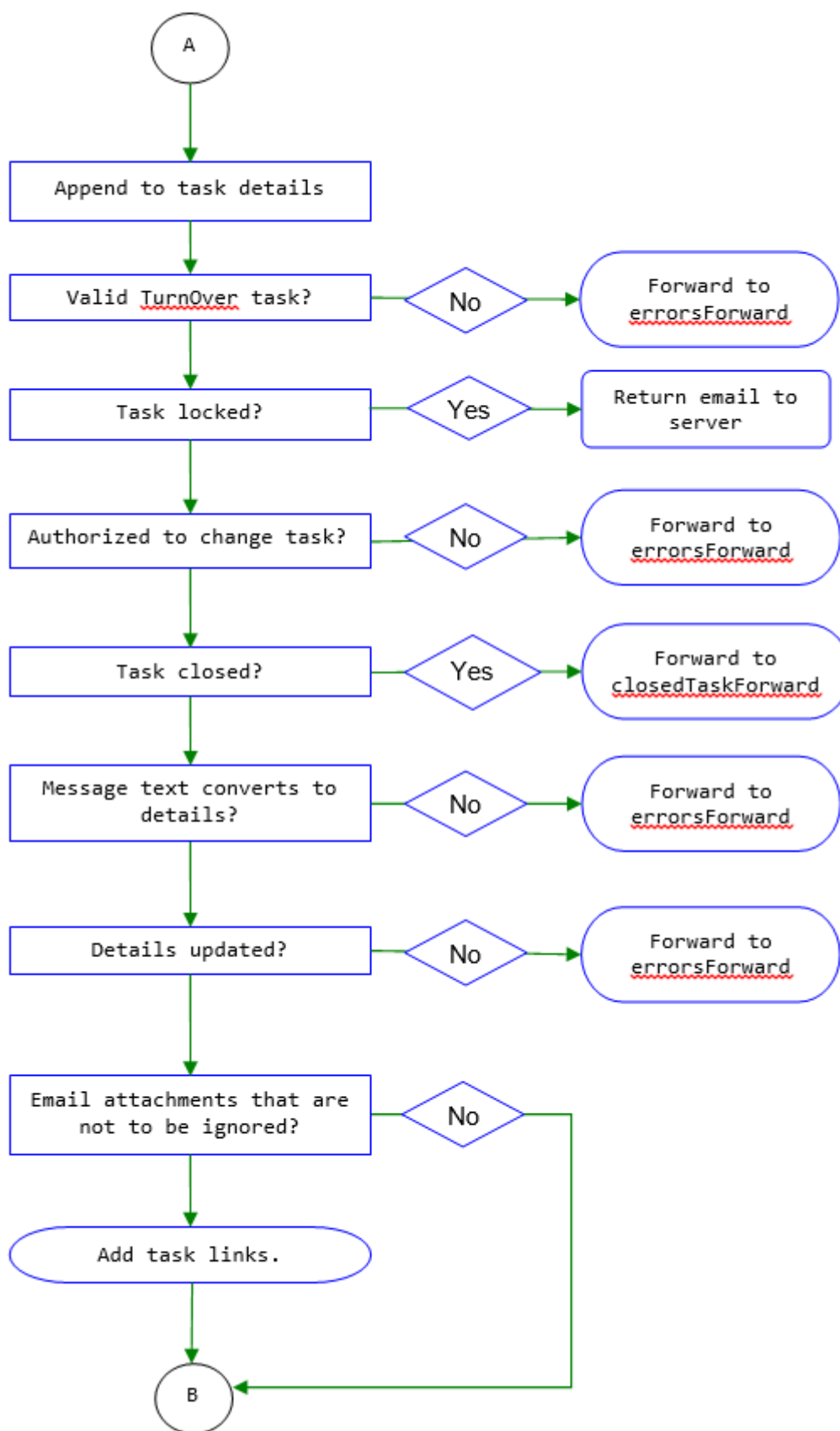
If you have multiple mailboxes and you only need to re-authenticate one of them you can edit this file and delete just the JSON object for the mailbox you need to re-authenticate. Be sure to remove the complete JSON object that has the same property value as the mailbox user you want to re-authenticate. This will prevent the other mailboxes from having to re-authenticate as well.

Specify the configuration properties file that contains the details of the remote IBM i server and TURNOVER installation libraries as the final parameter to the call.



## Appendix A – Flow diagram of mail process





# 18 Managing physical and logical files

This section describes how physical and logical files are managed in TURNOVER® Lifecycle Manager on the IBM i computer.

---

## New physical files

Physical File (PF) source code is created in a development library and then promoted into a target test or production library. The method of promotion can be copy source and compile object, move object or create duplicate object. The method is determined by value of the **Method** parameter of the application definition and, at run time, by the Method specified on the TURNOVER® Lifecycle Manager form line.

### File members

TURNOVER® Lifecycle Manager creates a new physical file with one member (member name equal to file name). You must add additional members to the file after the file is created using the `ADDPFM` command. TURNOVER® Lifecycle Manager does not automatically replicate members from the test file. The `ADDPFM` command can be entered as a post-run command of the same TURNOVER® Lifecycle Manager form.

### Data

Normally, a new file does not contain data; however, if data needs to be copied into the new file from the test file, you can change the `CPYFMT` parameter of the TURNOVER® Lifecycle Manager form line record to specify `*FROMFILE`. To do so, edit the form and change the line details.

If data needs to be copied from a file in some other location (a file in another library), you can include a post-run command to copy that file.

If data needs to be converted from another file (the new file is sufficiently different from the old), you can include a post-run command to call a conversion program to copy data to the new file. The conversion program can be turned over on the same form, if you want.

### Overriding Create Physical File (CRTPF) parameters

Create parameters can either be specified by referring to a reference object, a test object or, when replacing an existing file, the old object. Occasionally, you may want to specify, explicitly in the create command, the parameters to be used when creating a particular file.

## Changing defaults on the Create PF command

The CRTPF command associated with the object type PF can be changed within TURNOVER® Lifecycle Manager without affecting the version of the command in QSYS. To do so, select Main Menu option 8 and then option 4. Type 2 next to the object type you want to change. Prompt the create command and change the parameters you want to change.

This change applies to all applications defined to TURNOVER® Lifecycle Manager on your system.

## Setting create parameters using a reference object

Certain physical file parameter values, such as size and maximum number of members, can be assigned to the new file based on the old file, the test file or a reference file. Parameters set this way include any parameter you can change using a change physical file command. The **Create parameters** field value determines how these values are set when the file is promoted.

You can set up reference objects such as:

REFPF, REFLF, REFMSGF, REFPGM, REFCMD, REFDSPF, REFPRTF

in your production library, and then type REF\* in the **Create Reference Object** field for the form line. TURNOVER® Lifecycle Manager parses together "REF" and the object type code (for example, CLP) and then looks for that reference object. TURNOVER® Lifecycle Manager uses that as the reference object.

The Create parameter default values are as follows:

| Value | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| P=    | Use the target object as a reference for relevant parameters when creating this object. If the object is new, use the named reference object (REF* or equivalent), if present. If the reference object is specified but isn't found, an error is issued. If the <b>Reference object</b> isn't specified (that is, is blank), TURNOVER® Lifecycle Manager uses the create command defaults, as specified on the TURNOVER® Lifecycle Manager Object Type File command field as if this field had been coded S. If the application is a multi-level application, and <b>Create parameters</b> is P and a production object exists at a higher level, TURNOVER® Lifecycle Manager uses the production object as the reference object and sets the <b>Create parameters</b> code to R. |
| T=    | Use the from-object as a reference for parameters when creating this object. If the object doesn't exist in test, use the named reference object or the TURNOVER® Lifecycle Manager type code command default. This value cannot be used on the remote computer.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| S=    | Use the command in the TURNOVER® Lifecycle Manager Object Type file for the create object command.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| R=    | Use the named object (entered in the space provided on the panel) as a reference when creating this object. If the reference object does not exist, TURNOVER® Lifecycle Manager issues an error message.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

| Value | Description |
|-------|-------------|
|-------|-------------|

|    |                                                                       |
|----|-----------------------------------------------------------------------|
| C= | Parameters are specified explicitly when a line is added to the form. |
|----|-----------------------------------------------------------------------|

## Overriding file creation parameters

Occasionally, you may want to override physical file parameters specified for the create command or reference object (see '[Overriding Create Physical File \(CRTPF\) parameters](#)' on page 323).

You can change (override) the create physical file command parameters that you want to use. To do so, edit the TURNOVER® Lifecycle Manager form and select the form line containing the file using option 10, and then override the command parameters. Also, from the Programmer Worklist, you can use option 36 (Compile) and press F4 to prompt. TURNOVER® Lifecycle Manager remembers the parameter settings when it creates the object; and if you change the same file again later, TURNOVER® Lifecycle Manager prompts you to include the same overrides. For more information, see:

'Line defaults panel' in the *TURNOVER® Lifecycle Manager User Guide*.

'Changing a form line' in the *TURNOVER® Lifecycle Manager User Guide*.

## Change the file using a post-run command

You can include a post-run command to change or manipulate a file after it has been created by TURNOVER® Lifecycle Manager. To enter a post-run command, edit the TURNOVER® Lifecycle Manager form and type 12 on the form line for your file. Additionally, you can use option 38 (Pre/post-Cmds) from your Programmer Worklist. Type the post-run command. TURNOVER® Lifecycle Manager associates this command with the file. If you change it again later, TURNOVER® Lifecycle Manager prompts you to include the same post-run command again.

## Replacing physical files

To checkout out source for a physical file, TURNOVER® Lifecycle Manager refers to the object description and application definition to locate the source, and then copies it to your development library.

If you use TURNOVER® Lifecycle Manager cross-reference, or a product interfaced to by TURNOVER® Lifecycle Manager (Hawkeye's Pathfinder or HelpSystems' Abstract), you can consult the cross-reference to display a list of related objects that refer to the file you are checking out. If you need to change any of these, check out the related object. If you only need to recompile the object later, do not check it out; TURNOVER® Lifecycle Manager enables you to recreate the dependent objects in production later, when you promote the file to production.

The programmer can make changes to the file, compile the object, and test the file in the development library.

## Create and run TURNOVER® Lifecycle Manager form (Promotion)

When you add a physical file to a TURNOVER® Lifecycle Manager form, TURNOVER® Lifecycle Manager checks for logical files related to the physical file in the target library (the file being replaced). If logical files exist, TURNOVER® Lifecycle Manager adds these to the form automatically. TURNOVER® Lifecycle Manager *does not allow a form to be submitted to run which does not have related logical files on the form*. If you do not want to recreate the logical files, you must include lines on the form to delete the logicals you don't want to recreate.

### Promotion

You can compile, move, or duplicate files into the target library.

When the TURNOVER® Lifecycle Manager form runs, TURNOVER® Lifecycle Manager first creates an archive library and moves the old physical file from the target library into the archive library. Then it creates or moves the new physical file into the target library. During the final phase of the promotion job, it copies data from the old file in the archive library to the new file. If, for any reason, the form does not complete successfully, TURNOVER® Lifecycle Manager deletes the new file and moves the old file back into the target library.

If a file is being journaled, TURNOVER® Lifecycle Manager ends journaling before attempting to replace it. Once the file has been replaced, and the data has been copied, TURNOVER® Lifecycle Manager resumes journaling.

If any problems are encountered during the promotion run, or if your applications **Archive objects** parameter is set **Yes**, TURNOVER® Lifecycle Manager keeps the archive library on the system, otherwise it deletes the archive library (and the old physical file) at the end of the TURNOVER® Lifecycle Manager job.

### Data

When a file is replaced, the data is normally copied from the old file of the target library to the new file, member by member using the copy command with format option of **\*MAP \*DROP** (the normal default). Alternatively, you can specify copy format parameters of **\*NOCHK** (no check, no tell), **\*NONE** (don't copy the data), or **\*FROMFILE** (the data is copied from the from file to the new file). The default copy method is determined from the application definition parameter **Field mapping default for copy file command**. You can override this on the TURNOVER® Lifecycle Manager form by editing the form, selecting the physical file line with option 2, and changing the **FMTOPT** parameter.

If TURNOVER® Lifecycle Manager cannot successfully copy the data from the old file to the new one, the form ends with warning messages and retains the old copy of the file in the archive library.

### More information

For more information about how TURNOVER® Lifecycle Manager manages the data conversion of a file during promotion, see:

- ['Using the CHGPF command' on page 332](#)

- [‘Change process’ on page 52](#)
- *TURNOVER® Data Manager User Guide.*

## File members

If a multiple-membered file is being replaced, TURNOVER® Lifecycle Manager recreates the file with all the members that were in the physical file.

If you need to add or remove members to the file, you must do so using a post-run command. Edit the form and select the file line with option 12 and then use the `ADDPFM` or `RMVM` command.

---

## Overriding Create Physical File (CRTPF) parameters

The same rules apply when replacing files as when creating new files: see [‘New physical files’ on page 319](#).

---

## Deleting physical files

To delete a physical file, add a line to a form for the object and specify an action code of `D` (delete).

After you have completed editing the form, TURNOVER® Lifecycle Manager inserts lines on the form to delete all related logical files for the physical file being deleted.

---

## New logical files

Logical File (LF) source code is created in a development library and then promoted into target test or production libraries. The method of promotion is always copy source and compile the object, no matter what the application definition default setting in the line defaults. The method can be overridden in the LF type code, but it is usually best to allow the default method of `CSCO` to be used for logicals.

## File members

Normally, TURNOVER® Lifecycle Manager creates the logical file with one member (member name equal to file name), however, you can create or add additional members in several ways. TURNOVER® Lifecycle Manager does not automatically replicate members from the test file.

You can add additional members to the file after the file is created using a post-run command. Edit the form, position to the last line of the form, press `F18=Commands`, and type the `ADDLFM` command.

You can override the create logical file command on the TURNOVER® Lifecycle Manager form. Edit the TURNOVER® Lifecycle Manager form, select the logical file line with a 10, and enter the appropriate *physical file data member* parameters.

## Data

If data exists in the related physical files, a logical file index is built at the time the logical file is created. If the physical file and logical files are on the same form, then they are created with no data.

---

## Overriding Create Logical File (CRTLF) parameters

Create parameters can be specified by referring to a reference object, a test object or, when replacing an existing logical file, the old object. Occasionally, you may want to specify explicitly the parameters to be used when creating a particular logical file.

## Changing defaults on the Create LF command

The `CRTLF` command associated with the object type LF can be changed within TURNOVER® Lifecycle Manager without affecting the version in the command in QSYS.

This change applies to all applications defined to TURNOVER® Lifecycle Manager on your system.

- 1 Select Main Menu option 8 and then option 4.
- 2 Type 2 next to the object type you want to change.
- 3 Prompt the create command and change the parameters you want to change.

## Setting create parameters using a reference object

Certain logical file parameter values can be assigned to the new file based on the old file, the test file, or a reference file. Parameters set this way include any parameter you can change using a change logical file command. The **Create parameters** field value determines how these values are set when the file is promoted.

You can set up reference objects such as:

`REFLF, REFPF, REFMSGF, REFPGM, REFCMD, REFDSPF, REFPRTF . . .`

in your production library, and then type `REF*` in the **Create Reference Object** field for the form line. TURNOVER® Lifecycle Manager parses together "REF" and the object type code (for example, CLP) and then looks for that reference object. TURNOVER® Lifecycle Manager uses that as the reference object.



## Create parameter default values

| Value | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| P     | <p>Use the target object as a reference for relevant parameters when creating this object.</p> <p>If the object is new, use the named reference object (<code>REF*</code> or equivalent), if present. If the reference object is specified but is not found, an error is issued.</p> <p>If the <b>Reference object</b> is not specified (that is, it is blank), TURNOVER® Lifecycle Manager uses the create command defaults, as specified on the Object Type File command field as if this field had been coded <code>S</code>.</p> <p>If the application is a multi-level application, and <b>Create parameters</b> is <code>P</code> and a production object exists at a higher level, TURNOVER® Lifecycle Manager uses the production object as the reference object and sets the <b>Create parameters</b> code to <code>R</code>.</p> |
| T     | <p>Use the from-object as a reference for parameters when creating this object. If the object does not exist in test, use the named reference object or the TURNOVER® Lifecycle Manager type command defaults.</p> <p>This value cannot be used on the remote computer.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| S     | <p>Use the command in the TURNOVER® Lifecycle Manager Object Type file for the create object command.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| R     | <p>Use the named object (entered in the space provided on the panel) as a reference when creating this object.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| C     | <p>Parameters are specified explicitly when a line is added to form.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

## Overriding file creation parameters

You can override logical file parameters specified for the create command or reference object (see *'Overriding Create Logical File (CRTL) parameters' on page 324*).

To do so, do one of the following:

- Edit the TURNOVER® Lifecycle Manager form and select the form line containing the file using option 10, and then override the command parameters.
- From the Programmer Worklist, use option 36 (Compile), and then press F4 to prompt.

TURNOVER® Lifecycle Manager remembers the parameter settings when it creates the object; and if you change the same file again later, TURNOVER® Lifecycle Manager prompts you to include the same overrides.

### More information

'Line defaults panel' in the *TURNOVER® Lifecycle Manager User Guide*.

'Changing a form line' in the *TURNOVER® Lifecycle Manager User Guide*.

---

## Change the file using a post-run command

You can include a post-run command to change or manipulate a file after it has been created by TURNOVER® Lifecycle Manager.

To enter a post-run command, do one of the following:

- Edit the TURNOVER® Lifecycle Manager form and type a 12 on the form line for your file. Type the post-run command.
- Use option 38 (Pre/post-Cmds) from your Programmer Worklist. Type the post-run command.

TURNOVER® Lifecycle Manager associates this command with the file. If you change it again later, TURNOVER® Lifecycle Manager prompts you to include the same post-run command again.

---

## Replacing logical files

To checkout out source for a logical file, TURNOVER® Lifecycle Manager refers to the object description and the application definition to locate the source, and then copies it to your development library.

If you use TURNOVER® Lifecycle Manager's cross reference, Hawkeye's Pathfinder or HelpSystems' Abstract you can consult the cross reference to display a list of related objects that refer to the file you are checking out. You can check out the related objects. If you only need to recompile the object later, then do not check it out; TURNOVER® Lifecycle Manager enables you to recreate the dependent objects in production later, when you promote the file to production.

---

## Create and run TURNOVER® Lifecycle Manager form (Promotion)

When you add a physical file to a TURNOVER® Lifecycle Manager form, TURNOVER® Lifecycle Manager checks for logical files related to the physical file in the target library (the file being replaced). If logical files exist, TURNOVER® Lifecycle Manager adds these to the form automatically. TURNOVER® Lifecycle Manager *does not allow a form to be submitted to run that does not have related logical files on the form*. If you do not want to recreate the logical files, you must include lines on the form to delete the logicals you do not want to recreate.

### Promotion

Logical files are always compiled in TURNOVER® Lifecycle Manager.

For exceptions to this rule, see *'Why TURNOVER® Lifecycle Manager always compiles logical files' on page 328*.

When the TURNOVER® Lifecycle Manager form runs, TURNOVER® Lifecycle Manager creates an archive library and moves the old logical file from the target library into the archive library. Then it creates or moves the new logical file in the target library.

If you replace a physical and logical file on the same form, the physical file and related logical files are all in the archive library. The new PFs and LFs are created in the target library.

If you replace only a logical file, the related physical file has the old and new access paths associated with it until you remove the archive library. This method facilitates rapid recovery, if you decide to replace the new file with the old one.

If any problems occur during the promotion run, or if the application's **Archive objects** parameter is set `Yes`, TURNOVER® Lifecycle Manager keeps the archive library on the system; otherwise, it deletes the archive library and the old logical file at the end of the job.

## Data

When a logical file is replaced, if the physical file is populated with data, the access path is built when the logical file object is created.

## File members

If a multiple-membered logical file is being replaced, TURNOVER® Lifecycle Manager recreates the file with all the members that were in the original logical file.

To add or remove members or restructure the logical files member relationships to the related physical file members, use a post-run command. Edit the form position the last line on the form, press F18, and then use the `ADDLFM` or `RMVM` command.

## Overriding Create Logical File (CRTLF) parameters

The same rules apply when replacing files as when creating new files.

---

# Deleting logical files

To delete a logical file, add a line to a form for the object and specify an action code of `D` (delete).

---

# Duplicating files into Explode libraries

Explode libraries can be defined where you have an application which manages multiple divisions, companies, or a similar organizational entity. This feature enables you to create a data library for each organizational entity and maintain them in parallel. When TURNOVER® Lifecycle Manager promotes a data object to the primary library, it also duplicates that object into each of the explode libraries.

Physical files are always duplicated into Explode libraries, preserving the members in the file being replaced (if applicable).

Logical files are usually *compiled* into explode libraries. However, if the promotion method is CSCD, CSMO, CD, or MO, logical files are duplicated into the explode libraries. Before processing a logical file, TURNOVER® Lifecycle Manager puts the explode library into the library list (\*FIRST), and then performs the compile/duplication. If you have overridden the compile command parameters, then TURNOVER® Lifecycle Manager uses the overridden version of the command when creating the logical file. You can also define the library list to be used for each explode create operation, if necessary.

If reference (REF\*) objects exist in an Explode library, TURNOVER® Lifecycle Manager uses the reference object for assigning create parameters and authority, if the **Create Parm** and **Authority** parameters on the TURNOVER® Lifecycle Manager form line are set to R.

## Using post-run commands with objects in Explode libraries

If you use the explode library feature, any post-run command which contains "&ALLTGT" (uppercase, including the quotes) in a library parameter are applied repetitively to the primary target library and to each explode library.

---

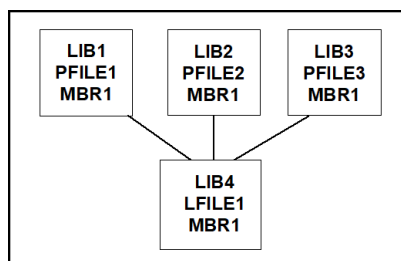
## Why TURNOVER® Lifecycle Manager always compiles logical files

To avoid conditions that might result in logical files being created over the wrong physical files, we always create logical files from source, using the CRTLF command.

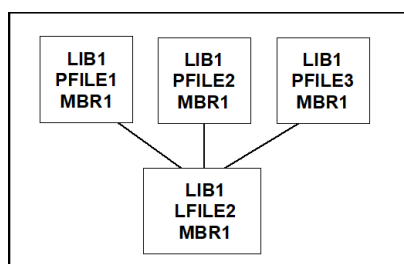
### Logical files and members

Logical files can contain up to 32 members. LF members can be over one PF member or multiple PF members. LF members can be over multiple PF members held in one or more PFs in one or more libraries. Join LF members can be over multiple PF members. TURNOVER® Lifecycle Manager can handle all of these scenarios; all of the following are legitimate:

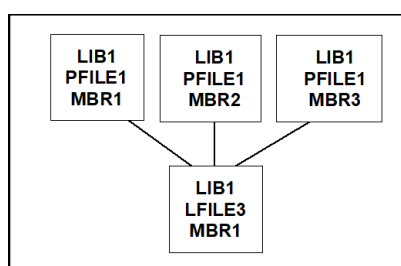
- 1 Logical file joins data from multiple PFs in multiple libraries. Each PF has a different format.



- 2 Logical file joins data from multiple PFs in the same library. Each PF has a different format.



- 3 Logical file is over data from multiple members of the same library. Each PF has the same format.



## Processing

### Replacing logical files

When replacing LFs, TURNOVER® Lifecycle Manager preserves members just as they are in the target library. The developer must use a pre- or post-run command (or other methods described earlier) to add or remove members.

### Creating new logical files

When adding new LFs, the standard default is to create a logical file with one member of the same name as the file itself. If additional/different members are desired, then use the post-run command to add or remove members.

### User methods other than CSCO

Where logical files are concerned, methods `MO` (Move Object) and `CD` (Create Duplicate) are always converted to `CSCO` (Copy Source, Compile Object) to ensure that the *access path is rebuilt over the target physical file(s)*. If the LF in the from-library is based on a PF in a different library, or if the LF in the target-library is to be based on a PF in a different library, methods `MO` or `CD` could result in errors; therefore, TURNOVER® Lifecycle Manager always uses `CSCO` to ensure that the logical is compiled over the correct physical as determined by the library list when the TURNOVER® Lifecycle Manager form is run. If the physical and logical are in the same from-library, and the physical and logical are in the same target library, then method `CD` would work; however, for consistency, create object is always used.

## Using “Explode” libraries

When changing PFs, the changed PF is duplicated from the base target library to one or more “explode” libraries. The members and their data are preserved. The data is copied from the old version of the file to the new version within the explode library, member by member, using the CPYF command with \*MAP \*DROP (the normal default), \*NOCHK, \*FROMFILE, or \*NONE (whatever was specified for the primary library on the TURNOVER® Lifecycle Manager form line).

When changing a logical file where explode libraries are used, TURNOVER® Lifecycle Manager always compiles the logical file into the explode library using the create command specified for the TURNOVER® Lifecycle Manager form line. Before each compile, TURNOVER® Lifecycle Manager adds the target explode library to the library list in the first position. If specific libraries are specified for each explode library entry, then the library list is replaced with that library list before running each compile.

## Exception

If the method is specified in the type code, then that method is used.

---

## Distributing physical and logical files

Object distribution occurs after successful promotion of objects to the application level from which you want to distribute. Source, object or both can be distributed, depending upon settings in the System, Application and Application type code entries (source is always distributed for logical files).

If source is distributed and no source file exists in the target source library, then TURNOVER® Lifecycle Manager creates the appropriately named source file.

If no source library is identified in the application definition, then TURNOVER® Lifecycle Manager uses QTEMP. You are recommended to create and identify a source library on each target computer. The target source library may be your data library.

Whether or not you compile physical files on the development computer is determined by whether you distributed source, objects or both and, also, the setting of the Method parameter on the Application definition on the target system.

If you have overridden the create command on the TURNOVER® Lifecycle Manager form, TURNOVER® Lifecycle Manager distributes the create command with the distribution and uses the overrides on the target computer; otherwise TURNOVER® Lifecycle Manager uses the create command for the PF or LF type that is defined in the global type code entry on the target computer.

Physical files are normally distributed without data. If you specify the TURNOVER® Lifecycle Manager form line parameter CPYFMT of \*FROMFILE, then the data is also distributed and is promoted with the object.

Logical files are always compiled (unless the compilation is inhibited by the LF type code entry being set to **Source file name** to `NONE` or a method is specified in the type code). Logical file source is always distributed to all target computers. For this reason, you need to be sure that the application job description has an appropriate library list defined in order to create the logical file correctly.

If, for whatever reason, TURNOVER® Lifecycle Manager cannot locate the source on the target computer when recreating a logical file, then it duplicates the logical file object from the distribution library to the target library.

---

## Exception conditions

### Removing a key from a join logical file key

When you remove a key from a join logical file definition, you must delete the logical using one TURNOVER® Lifecycle Manager form and recreate the new logical on a separate form, to avoid conflicts.

You should run the form that deletes the logical with **Archive Objects** = `N`. If the delete form runs with **Archive Object** = `Y`, you get a warning on the TURNOVER® Lifecycle Manager log if the logical is built over a physical file in a production library.

## 19 Using the CHGPF command

TURNOVER® Lifecycle Manager can promote a physical file using the `CHGPF` command, instead of a more “normal” promotion method such as `CSCO` and then somehow mapping in the old data from the T#D library.

### What is the CHGPF command?

The Change Physical File (`CHGPF`) command permits a simple data conversion. `CHGPF` can be useful if you are just adding a new field to the end of a file because the OS does some low-level processing to preserve the existing logical file access paths, which makes it all happen faster.

The `CHGPF` command merges these separate steps into one:

- Create the new physical file from the source code indicated in the `CHGPF` command.
- Rebuild all logical files that are attached to the new physical file – providing they share the same record format as the physical file.
- Copy the data from the old physical file into the new physical file, using the `CPYF` command with a `*MAP *DROP` option.
- Delete the old physical file and its logicals.
- Move the new physical file and its logicals back into place.

### Limitations for using CHGPF

There are several scenarios where `CHGPF` either does not work well, or does not work at all:

- `CHGPF` only does a `*MAP *DROP`, so if you need conversion logic to change the fields (such as computing the century of a `MMDDYY` date), you can’t use `CHGPF`.
- `CHGPF` simply rebuilds the existing access paths as they are. If you wanted them to pick up the new field, you would have to handle that in a more “normal” fashion.
- If your logical files do not actually define the individual fields and instead simply pull in the record of the physical file, then it would appear confusing. Someone looking at the actual logical file would wonder why it is not really showing all the fields in the physical file.
- You must have the source code available on the machine during the conversion (this might be a problem on remote or production boxes).



- The physical file object has to be in place. This may be a problem if you have the CHGPF command on a form that is running up to a QA level, and the physical file does not exist there. (UNICOM Systems, Inc. recommends keeping a full complement of physical and logical files at the QA level, even if they contain a subset of records.)

## Advantages

The biggest gain is that you do not have the extra downtime associated with a normal data conversion, especially when it comes to rebuilding the access paths. There is some downtime, but not as much, mostly because of the time saved in the access path rebuilds.

## Disadvantages

- If you use this method, you lose some of the fundamental strengths that TURNOVER® Lifecycle Manager provides. TURNOVER® Lifecycle Manager cannot build and run a recovery form to restore files changed with the CHGPF type code.
- TURNOVER® Lifecycle Manager can not help you maintain company standards, such as object owner and access authorities (very important on a physical file).
- You can not use worklist option 36=Compile at the development level because this is a source-only object. You could use a two position user-defined option or the command line to compile it. However, TURNOVER® Lifecycle Manager does not be helping you with default create command parameters, the way it was compiled last time, ownerships, and so on.
- You must manually override TURNOVER® Lifecycle Manager's automatic impulse to add the logical files as recompile-only objects on each form.
- You can not use TURNOVER® Lifecycle Manager recovery forms to restore objects that have been archived using the CHGPF process.

## Other options

If you prefer to have all of the benefits of a more normal promotion (such as archiving, recoverability, and so on), but you cannot afford the downtime, consider TURNOVER® Data Manager.

TURNOVER® Data Manager can perform data conversion, including the access path builds, while users are still signed on. It does all the data conversion work in a side staging file. The only downtime that occurs happens at the end of the process, when TURNOVER® Data Manager gets an exclusive lock on the file, moves the old file out, and replaces it with the new file.

TURNOVER® Lifecycle Manager invokes TURNOVER® Data Manager during form processing.

---

## Setting up CHGPF

For the CHGPF technique to work, you must tell TURNOVER® Lifecycle Manager to do some special processing as it promotes these types of objects. To accomplish this, use the TCHGPF change management scheme.

## Restore the TCHGPF change management scheme

- 1 Locate the TCHGPF save file in the TURNOVER® Lifecycle Manager product library SOFTTURN (substitute in your product library name if different).

You can also download it from <https://support.softlanding.com>

- 2 From a TURNOVER® Lifecycle Manager command line, prompt the TWRKCMSCH command.
- 3 Press F8: Restore. Fill out the parms as follows, substituting library SOFTTURN with your TURNOVER® Lifecycle Manager product library if necessary:

If you found the save file already on your computer:

```
SAVF(SOFTTURN/TCHGPF)  TOScheme(*SAVScheme)  REPLACE(*NO)  RSTPGMS(*NO)
RSTRSRCLIB(*NO)
```

If you downloaded the save file from the support site:

```
SAVF(SOFTTURN/TCHGPF)  TOScheme(*SAVScheme)  REPLACE(*NO)  RSTPGMS(*YES)
PGMLIB(*Scheme)  RSTRSRCLIB(*YES)  RSRCLIB(*SAVLIB) .
```

- 4 If your product library is not SOFTTURN, you must change where the scheme should look for the programs that run any methods pointing to \*DEFAULT. Select TCHGPF with option 2 and complete the parameters as follows:

```
DFTPGM(yourlib/*TURNOVER)
```

Replace *yourlib* with the name of the TURNOVER® Lifecycle Manager product library.

## Create a new type code

Create a new type code to tell TURNOVER® Lifecycle Manager how to handle the unique requirements. Do the following:

- 1 Copy the physical file type code to create a new CHGPF type code.
- 2 Edit the lines as shown here:

|                                                                              |                                                                                       |                                        |                               |
|------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|----------------------------------------|-------------------------------|
| MM/DD/YY<br>10:32:53                                                         | <b>TurnOver Change a Type Code</b>                                                    |                                        | YOUR COMPANY, INC.<br>YOURSYS |
| Type code . . . . .                                                          | <u>CHGPF</u>                                                                          | CM scheme . . . . .                    | <u>TCHGPF</u>                 |
| Description . . . . .                                                        | <u>Promote file with CHGPF</u>                                                        | Attribute . . . . .                    | <u>TCHGPF</u>                 |
| Object type . . . . .                                                        | <u>*FILE</u>                                                                          | Source file name . .                   | <u>QDDSSRC</u>                |
| Sequence . . . . .                                                           | <u>410</u>                                                                            | Data object . . . . .                  | <u>Y COPR</u>                 |
| Create command . . .                                                         | <u>CHGPF FILE("&amp;LI"/"&amp;OB") SRCFILE("&amp;SL"/"&amp;SF") SRCMBR("&amp;SM")</u> |                                        |                               |
| <u>SIZE(*NOMAX)</u>                                                          |                                                                                       |                                        |                               |
|                                                                              |                                                                                       |                                        |                               |
| &OB" = Object name                                                           |                                                                                       | &RF" = Reference      &X0"-&X9" = Exit |                               |
| &LI" = Library name                                                          |                                                                                       | &TO" = Test object name                |                               |
| &TY" = Type code                                                             |                                                                                       | &TL" = Test object library name        |                               |
| &SM" = Source member name                                                    |                                                                                       | &TR" = Target release                  |                               |
| &SF" = Source file name                                                      |                                                                                       | &FM" = From source member name         |                               |
| &SL" = Source library name                                                   |                                                                                       | &FF" = From source file name           |                               |
| &U0" = Generation options                                                    |                                                                                       | &FL" = From source library name        |                               |
| &U2" = ILE Debugging View                                                    |                                                                                       | &U1" = Product library (commands)      |                               |
| &U4" = PMTFILE Message File                                                  |                                                                                       | &U3" = ILE Optimization Level          |                               |
| &U6" = ILE Compiler Options                                                  |                                                                                       | &U5" = Sort Sequence                   |                               |
| &U8" = LF Parent, Lib, Members                                               |                                                                                       | &U7" = Activation Group                |                               |
|                                                                              |                                                                                       | &U9" = ILE Compiler Options            |                               |
| F3=Exit   F4=Prompt/select   F7=Applications   F12=Cancel   F22=Long command |                                                                                       |                                        |                               |

- 3 Move the cursor to **CM Scheme**, press F4, and then select TCHGPF.

| Parameter              | Value                                                         |
|------------------------|---------------------------------------------------------------|
| Description            | Promote file with CHGPF                                       |
| Attribute              | TCHGPF                                                        |
| Data object and method | Y COPR                                                        |
| Create command         | CHGPF FILE("&LI"/"&OB") SRCFILE("&SL"/"&SF")<br>SRCMBR("&SM") |

- 4 Press F7, and then push this type code into all levels of all desired applications.
- 5 Exit and save the new type code.
- 6 If you want to distribute TURNOVER® Lifecycle Manager forms containing this new type code to remote computers, repeat this process on each of those remote computers.

## Change and promote the physical file

- 1 Add the physical file to the worklist, specifying the CHGPF type code.
- 2 Check out the object, and then edit the source code.

Worklist option 36 defaults to the CHGPF command, which is *not* what you want here at development. Either use the standard CRTPF command on your command line, or create a two-position worklist short cut to create and unit test the new file.

- 3 Build the form at the next level, and then edit the form.

TURNOVER® Lifecycle Manager automatically adds to the form any recompile-only objects it finds in the cross-reference. This is suitable for programs, but it is probably not what you want for logical files in this case, because the CHGPF command handles this step already. Delete the logical files that TURNOVER® Lifecycle Manager added automatically from the form.

- 4 Exit and save the form. Remember to change the “Check cross-reference” line to “No”; if you do not, TURNOVER® Lifecycle Manager add all of the items that you just deleted again.
- 5 Run the form.
  - The IBM CHGPF command is not replacing the original object, so you lose the TURNOVER® Lifecycle Manager advantage of archiving (and recovering) the original file and its data. However, the TCHGPF CM Scheme does duplicate the existing file, with its data, into the T-Form #-D library.
  - Normal source archiving also occurs, if that option is configured for your application.
- 6 If you have additional levels above this one, you can copy the form. TURNOVER® Lifecycle Manager does not re-add the logicals from the cross-reference unless you open the form in edit mode and then forget to turn off the “check cross-reference” switch on the way out.

## 20 Handling message IDs

You can control and track changes to individual message IDs in TURNOVER<sup>®</sup> Lifecycle Manager. Each message ID can be checked out, changed and promoted independently, without having to lock or merge a message file.

### Defining type codes for message IDs

You change message IDs with TURNOVER<sup>®</sup> Lifecycle Manager by creating a special type code for handling the messages in a particular message file. The key component to the type code is the object type parameter that must be \*MSGID. The type code, with the exception of the name and description you assign, should look like this:

| MM/DD/YY                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                       | Change a Type Code     |           | Your Company, Inc. |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                          |                                       |                             |                           |                           |                             |                                                                      |  |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------|------------------------|-----------|--------------------|--|-------------------|--------------------|---------------------|--------------------------|----------------------|----------------------------------|-------------------|------------------------|----------------------------|---------------------------------|--------------------------|-------------------------------|-----------------------------|----------------------------------|----------------------------|-------------------------|----------------------------|--------------------------|--------------------------|---------------------------------------|-----------------------------|---------------------------|---------------------------|-----------------------------|----------------------------------------------------------------------|--|
| 09:41:45                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                       |                        |           | YOURSYS            |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                          |                                       |                             |                           |                           |                             |                                                                      |  |
| Type code . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | MSGID                                 | CM scheme . . . . .    | *TURNOVER |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                          |                                       |                             |                           |                           |                             |                                                                      |  |
| Description . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Message ID                            | Attribute . . . . .    |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                          |                                       |                             |                           |                           |                             |                                                                      |  |
| Object type . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | *MSGID                                | Source file name . . . |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                          |                                       |                             |                           |                           |                             |                                                                      |  |
| Sequence . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 100                                   | Data object . . . . .  | Y COPR    |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                          |                                       |                             |                           |                           |                             |                                                                      |  |
| Create command . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                       |                        |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                          |                                       |                             |                           |                           |                             |                                                                      |  |
| <table border="0"> <tr> <td>"&amp;RF" = Reference</td> <td>"&amp;X0"-"&amp;X9" = Exit</td> </tr> <tr> <td>"&amp;OB" = Object name</td> <td>"&amp;TO" = Test object name</td> </tr> <tr> <td>"&amp;LI" = Library name</td> <td>"&amp;TL" = Test object library name</td> </tr> <tr> <td>"&amp;TY" = Type code</td> <td>"&amp;TR" = Target release</td> </tr> <tr> <td>"&amp;SM" = Source member name</td> <td>"&amp;FM" = From source member name</td> </tr> <tr> <td>"&amp;SF" = Source file name</td> <td>"&amp;FF" = From source file name</td> </tr> <tr> <td>"&amp;SL" = Source library name</td> <td>"&amp;FL" = From source library name</td> </tr> <tr> <td>"&amp;U0" = Generation options</td> <td>"&amp;U1" = Debugging views</td> </tr> <tr> <td>"&amp;U2" = Optimization level</td> <td>"&amp;U3" = Compiler options</td> </tr> <tr> <td>"&amp;U4" = Activation Group</td> <td>"&amp;U5" = WEBLICATION System designator</td> </tr> <tr> <td>"&amp;U6" = WEBLICATION Release</td> <td>"&amp;U7" = Binding directory</td> </tr> <tr> <td>"&amp;U8" = Binding directory</td> <td>"&amp;U9" = Run as User Profile</td> </tr> <tr> <td colspan="2">F3=Exit F4=Prompt/select F7=Applications F12=Cancel F22=Long command</td> </tr> </table> |                                       |                        |           |                    |  | "&RF" = Reference | "&X0"-"&X9" = Exit | "&OB" = Object name | "&TO" = Test object name | "&LI" = Library name | "&TL" = Test object library name | "&TY" = Type code | "&TR" = Target release | "&SM" = Source member name | "&FM" = From source member name | "&SF" = Source file name | "&FF" = From source file name | "&SL" = Source library name | "&FL" = From source library name | "&U0" = Generation options | "&U1" = Debugging views | "&U2" = Optimization level | "&U3" = Compiler options | "&U4" = Activation Group | "&U5" = WEBLICATION System designator | "&U6" = WEBLICATION Release | "&U7" = Binding directory | "&U8" = Binding directory | "&U9" = Run as User Profile | F3=Exit F4=Prompt/select F7=Applications F12=Cancel F22=Long command |  |
| "&RF" = Reference                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | "&X0"-"&X9" = Exit                    |                        |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                          |                                       |                             |                           |                           |                             |                                                                      |  |
| "&OB" = Object name                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | "&TO" = Test object name              |                        |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                          |                                       |                             |                           |                           |                             |                                                                      |  |
| "&LI" = Library name                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | "&TL" = Test object library name      |                        |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                          |                                       |                             |                           |                           |                             |                                                                      |  |
| "&TY" = Type code                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | "&TR" = Target release                |                        |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                          |                                       |                             |                           |                           |                             |                                                                      |  |
| "&SM" = Source member name                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | "&FM" = From source member name       |                        |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                          |                                       |                             |                           |                           |                             |                                                                      |  |
| "&SF" = Source file name                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | "&FF" = From source file name         |                        |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                          |                                       |                             |                           |                           |                             |                                                                      |  |
| "&SL" = Source library name                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | "&FL" = From source library name      |                        |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                          |                                       |                             |                           |                           |                             |                                                                      |  |
| "&U0" = Generation options                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | "&U1" = Debugging views               |                        |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                          |                                       |                             |                           |                           |                             |                                                                      |  |
| "&U2" = Optimization level                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | "&U3" = Compiler options              |                        |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                          |                                       |                             |                           |                           |                             |                                                                      |  |
| "&U4" = Activation Group                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | "&U5" = WEBLICATION System designator |                        |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                          |                                       |                             |                           |                           |                             |                                                                      |  |
| "&U6" = WEBLICATION Release                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | "&U7" = Binding directory             |                        |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                          |                                       |                             |                           |                           |                             |                                                                      |  |
| "&U8" = Binding directory                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | "&U9" = Run as User Profile           |                        |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                          |                                       |                             |                           |                           |                             |                                                                      |  |
| F3=Exit F4=Prompt/select F7=Applications F12=Cancel F22=Long command                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                       |                        |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                          |                                       |                             |                           |                           |                             |                                                                      |  |

When you apply TURNOVER<sup>®</sup> Lifecycle Manager changes, one MSGID type entry will be created for you. If you need more, you can copy it.

You can use the same type code to control different message files in different applications.

If you need to manage more than one message file in a single application, you need a unique type code for each message file. Set the sequence value for your message ID to a value that will place it on forms before any object types that may use it for compiling (such as CMDs, DSPFs, and PRTFs). Fill out the rest of the fields as shown above.

## Assigning a message ID type code to an application

After adding the type code to the list of valid type codes for your application, you need to maintain that application type code to specify the message file where TURNOVER® Lifecycle Manager should look for the message IDs. This should look like the following panel:

| MM/DD/YY                             | MSGID Defaults         |                                           | Your Company, Inc.                     |  |
|--------------------------------------|------------------------|-------------------------------------------|----------------------------------------|--|
| 10:23:56                             | Application: <b>TO</b> | Rel: <b>3</b> Ver: <b>2</b> Lev: <b>2</b> | SYSTEM: YOURSYS                        |  |
| Type choices and press Enter.        |                        |                                           |                                        |  |
|                                      |                        |                                           | Field Attributes<br>R=Restrict, L=Lock |  |
| <b>From</b> message file . . . . .   | <u>TMSGF</u>           | Name                                      | L                                      |  |
| Library . . . . .                    | <u>SOFTTURNQA</u>      | Name, *PGMR                               | L                                      |  |
| <b>Target</b> message file . . . . . | <u>TMSGF</u>           | Name                                      | R                                      |  |
| Library . . . . .                    | <u>SOFTTURN</u>        | Name                                      | R                                      |  |
| Delete from message ID . . . . .     | <u>Y</u>               | Y, N                                      | R                                      |  |
| Distribute . . . . .                 | <u>N</u>               | Y, N                                      |                                        |  |
| F3=Exit F12=Cancel                   |                        |                                           |                                        |  |

On this panel, define the **From** and **Target message files** for message IDs that are promoted on forms for this level of the application. You can also specify whether you want to remove message IDs from the **From message file** as they are promoted, and whether you want them to distribute message IDs on forms intended for distribution.

Once these IDs are defined to your application, your software developers can check out and promote message IDs.

## Distributing message IDs

All TURNOVER® Lifecycle Manager features are supported for message IDs using normal TURNOVER® Lifecycle Manager procedures. Form builds, runs and distributions are performed in the same manner you would any other object. When distributions take place, only the message IDs on the form are distributed.

TURNOVER® Lifecycle Manager never deletes a message file; only individual message descriptions.

## Managing type code definitions

You can maintain TURNOVER® Lifecycle Manager's system-supplied TURNOVER® Lifecycle Manager type codes.

TURNOVER® Lifecycle Manager is shipped with most object types already pre-defined.

You can control default command parameters such as **Target release** and **Replace \*YES/\*NO** that TURNOVER® Lifecycle Manager uses when it creates an object in Production.

You can create your own object type, as when you need to create an object using a vendor-supplied create command.

You can create your own command to use in place of the system-supplied commands, if you have some special requirements, such as collating source members before compiling an object.

Be careful when you change, delete, or create object type code entries. These entries affect all applications on your system. Coordinate changes with all TURNOVER® Lifecycle Manager users, and test new object types thoroughly. Be sure you understand how TURNOVER® Lifecycle Manager uses these entries before changing them.

### To see a list of available type codes

On the Utility menu, select option 4.

|          |                         |                         |                       |     |                        |  |
|----------|-------------------------|-------------------------|-----------------------|-----|------------------------|--|
| MM/DD/YY |                         | Work with Type Codes    |                       |     | Your Company, Inc.     |  |
| 09:22:35 |                         |                         |                       |     | YOURSYS                |  |
| Filter:  |                         | Position to . . . _____ |                       |     |                        |  |
| 2=Change |                         | 3=Copy                  | 4=Delete              | Dta |                        |  |
| Type     | Description             | Obj. Type               | Obj                   | Seq | Create CMD             |  |
| — AAUDIT | ASSET Audit Trail       | *AAUDIT                 | N                     | 250 | SRCFILE("&SL"/"&SF") S |  |
| — ABPGM  | ASSET Batch Program     | *ABPGM                  | N                     | 350 | SRCFILE("&SL"/"&SF") S |  |
| — ADMDL  | ASSET Data Model        | *ADMDL                  | N                     | 300 | SRCFILE("&SL"/"&SF") S |  |
| — ADPGM  | ASSET Display Program   | *ADPGM                  | N                     | 350 | SRCFILE("&SL"/"&SF") S |  |
| — AFILE  | ASSET File              | *AFILE                  | N                     | 200 | SRCFILE("&SL"/"&SF") S |  |
| — AFLD   | ASSET Field             | *AFLD                   | N                     | 100 | SRCFILE("&SL"/"&SF") S |  |
| — ARPGM  | ASSET Report Program    | *ARPGM                  | N                     | 350 | SRCFILE("&SL"/"&SF") S |  |
| — ASHLP  | ASSET Help Text         | *ASHLP                  | N                     | 350 | SRCFILE("&SL"/"&SF") S |  |
| — ASUBR  | ASSET Action Subroutine | *ASUBR                  | N                     | 250 | SRCFILE("&SL"/"&SF") S |  |
| — BAS    | Basic/400 program       | *PGM                    | N                     | 500 | CRTBASPGM PGM("&LI"/"& |  |
| — BASCPY | /COPY BAS Member        | *SRC                    | N                     | 100 |                        |  |
| — BKPDF  | Manage PDF'd book files | *IFSOBJ                 | Y                     | 20  |                        |  |
| — BMP    | Windows Bitmap          | *IFSOBJ                 | N                     | 100 |                        |  |
| — BND    | ILE Binder Language     | *SRC                    | N                     | 100 |                        |  |
| — BNDDIR | ILE Binding Directory   | *BNDDIR                 | N                     | 480 |                        |  |
|          |                         |                         |                       |     | More . . .             |  |
| F3=Exit  |                         | F6=Add Type             | F8=User-defined parms |     | F24=More Keys          |  |

### To

filter the list

### Do this

Use the **Filter** fields. For more information, see the online help.

| To                                                                | Do this                                                                                              |
|-------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|
| view or change the details associated with a type code            | Use option 2.                                                                                        |
| copy a type code in order to create a new one                     | Use option 3.<br>All related commands are also copied.                                               |
| delete a type code                                                | Use option 4.                                                                                        |
| create a new type code                                            | Press F6.<br><br>See <i>'Adding or changing TURNOVER® Lifecycle Manager type codes' on page 340.</i> |
| add and update user-defined parameters that can apply to any type | Press F8.                                                                                            |

## Adding or changing TURNOVER® Lifecycle Manager type codes

You can add your own type codes, so you can create variations of standard types.

You can use type codes in your application default definitions. TURNOVER® Lifecycle Manager will use the command you specify.

1 Do one of the following:

- To add a type code, press F6.
- To change a type code, select it with option 2.

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                      |                               |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|-------------------------------|
| MM/DD/YY<br>09:25:15                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | <b>Add a Type Code</b>               | Your Company, Inc.<br>YOURSYS |
| Type code . . . . . _____                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | CM scheme . . . . . <u>*TURNOVER</u> |                               |
| Description . . . . . _____                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Attribute . . . . . _____            |                               |
| Object type . . . . . _____                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Source file name . . . _____         |                               |
| Sequence . . . . . _____                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Data object . . . . . <u>N</u>       |                               |
| Create command . . . _____                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                      |                               |
| <hr/> <div> <div> "&amp;OB" = Object name<br/> "&amp;LI" = Library name<br/> "&amp;TY" = Type code<br/> "&amp;SM" = Source member name<br/> "&amp;SF" = Source file name<br/> "&amp;SL" = Source library name<br/> "&amp;U0" = Generation options<br/> "&amp;U2" = Optimization level<br/> "&amp;U4" = Source listing options<br/> "&amp;U6" = Unassigned<br/> "&amp;U8" = Binding directory<br/> F3=Exit F4=Prompt/select F8=Add IFS type code F12=Cancel F22=Long command </div> <div> "&amp;RF" = Reference<br/> "&amp;TO" = Test object name<br/> "&amp;TL" = Test object library name<br/> "&amp;TR" = Target release<br/> "&amp;FM" = From source member name<br/> "&amp;FF" = From source file name<br/> "&amp;FL" = From source library name<br/> "&amp;U1" = Debugging views<br/> "&amp;U3" = Compiler options<br/> "&amp;U5" = Unassigned<br/> "&amp;U7" = Service program source file<br/> "&amp;U9" = Module source file </div> <div> "&amp;X0"-"&amp;X9" = Exit </div> </div> |                                      |                               |

2 Specify the values for the type code, and then press Enter.



## Type code

The name of a new code must start with a recognizable IBM i type code. For example, a different version of the create CL program command from the command associated with CLP, you could call the new type code type code, CLPNO.

Another example of a type that is already set up is PFREF. This can be used exclusively for creating a field reference file. It is defined with MEMBER(\*NONE) and LVLCHK(\*NO) and assigned a form sequence code that ensures that it is always listed first on the TURNOVER® Lifecycle Manager form (it needs to be created before physical and logical files are created).

If you define your own type entry, use the Attribute field to tell TURNOVER® Lifecycle Manager what the real attribute is. TURNOVER® Lifecycle Manager uses this code in Programmer's Worklist Manager (PWM) when you edit source so the correct editor is used. For example, the PFREF type referred to above should be defined with an attribute of PF.

Three categories of type codes are possible:

| Category                    | Description                                                                                                                                                                                                                          |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Source-only types           | For example, text source members and copy books (see TXT type).                                                                                                                                                                      |
| Object-only types           | These types are always duplicated or moved into production. The associated command has no source reference (see DTAARA type). Object-only types without a command specified can only be moved or duplicated from test to production. |
| Source-created object types | Any object created using a create command, or moved or duplicated into production along with the source member from which the object was created (see RPG type).                                                                     |

TURNOVER® Lifecycle Manager also recognizes special types related to some case tools.

Some restrictions apply to the TURNOVER® Lifecycle Manager methods permitted to each of the above.

| Method                                       | Source only | Source compiled | Object only |
|----------------------------------------------|-------------|-----------------|-------------|
| CD      Create Duplicate Object              |             | X               | X           |
| COPR    Compile/Create in Production         |             | X               | X           |
| CS      Copy Source                          | X           |                 |             |
| CSCD    Copy Source, Create Duplicate Object |             | X               |             |
| CSCO    Copy Source, Compile Object          |             | X               |             |
| CSMO    Copy Source, Move Object             |             | X               |             |

| Method |             | Source only | Source compiled | Object only |
|--------|-------------|-------------|-----------------|-------------|
| MO     | Move Object |             | X               | X           |
| NULL   | No action   | –           | –               | –           |

*Source-only* types have \*SRC in the Object type code field.

*Source-compiled* types have a reference to a source member in the designated create command.

*Object-only* types have no create command, or a create command that does not reference a source member.

For more information about how each object type is promoted by TURNOVER® Lifecycle Manager, see 'Change Process' in the *TURNOVER® Lifecycle Manager Administrator's Guide*.

## Sequence

The **Sequence** field is used when sorting objects on a form into the correct logical order. For example, all PFs precede LFs that precede DSPFs, and so on. If you enter objects on a form out of order, TURNOVER® Lifecycle Manager displays a warning message.

## Create command

To enter a command string longer than 256 characters (often necessary with use of substitution variables or when binding multiple ILE programs), press F22.

The **Create command** and default parameters entered here are used when creating an object or transferring this object type to production. The substitution parameters appearing below the command line can be used wherever appropriate in the command.

For example:

```
CRTCBLPGM PGM("&LI"/"&OB") SRCFILE("&SL"/"&SF") SRCMBR("&SM")
```

| Parameter | Description                        |
|-----------|------------------------------------|
| &LI       | Library name (target library name) |
| &OB       | Object name (program name)         |
| &TY       | Type code                          |
| &SL       | Source library name                |
| &SF       | Source file name                   |
| &SM       | Source member name                 |
| &TO       | Test object name                   |

| Parameter | Description                         |
|-----------|-------------------------------------|
| &TL       | Test object library name            |
| &TR       | Target release                      |
| &FM       | From source member name             |
| &FF       | From source file name               |
| &FL       | From source library name            |
| &U0-9     | User-defined parameters (see below) |

If the command line is blank and the object type (above) is not \*SRC, only TURNOVER® Lifecycle Manager methods MO or CD will be allowed for this object type.

(See “Resolving the create command” in the *TURNOVER® Lifecycle Manager User Guide*.)

Enter a command name, and then press F4 to prompt it.

When entering the & variables, surround each one with a double quote.

---

## Adding or changing a type code for a change management scheme

Change management schemes enable you to override almost any change management method for a user-defined object type, for example, moving, copying and deleting an object. After you create a change management scheme and override its methods, you must create new type codes, or change existing ones, to use the new scheme. TURNOVER® Lifecycle Manager includes several sample schemes with their type codes already created.

### To add a type code

- 1 Press F6.

| MM/DD/YY                      |       | Add a Type Code                    |           | Your Company, Inc.          |  |
|-------------------------------|-------|------------------------------------|-----------|-----------------------------|--|
| 09:25:15                      |       |                                    |           | YOURSYS                     |  |
| Type code . . . . .           | _____ | CM scheme . . . . .                | *TURNOVER |                             |  |
| Description . . . . .         | _____ | Attribute . . . . .                | _____     |                             |  |
| Object type . . . . .         | _____ | Source file name . . .             | _____     |                             |  |
| Sequence . . . . .            | _____ | Data object . . . . .              | N         |                             |  |
| Create command . . . .        | _____ |                                    |           |                             |  |
| _____                         |       |                                    |           |                             |  |
| "OB" = Object name            |       | "RF" = Reference                   |           | "X0"-"X9" = Exit            |  |
| "LI" = Library name           |       | "TO" = Test object name            |           |                             |  |
| "TY" = Type code              |       | "TL" = Test object library name    |           |                             |  |
| "SM" = Source member name     |       | "TR" = Target release              |           |                             |  |
| "SF" = Source file name       |       | "FM" = From source member name     |           |                             |  |
| "SL" = Source library name    |       | "FF" = From source file name       |           |                             |  |
| "U0" = Generation options     |       | "FL" = From source library name    |           |                             |  |
| "U2" = Optimization level     |       | "U1" = Debugging views             |           |                             |  |
| "U4" = Source listing options |       | "U3" = Compiler options            |           |                             |  |
| "U6" = Unassigned             |       | "U5" = Unassigned                  |           |                             |  |
| "U8" = Binding directory      |       | "U7" = Service program source file |           |                             |  |
| F3=Exit F4=Prompt/select      |       | F8=Add IFS type code               |           | F12=Cancel F22=Long command |  |

2 In the **CM scheme** field, do one of the following:

- Type the name of the change management scheme.
- To select a change management scheme from a list, press F4.

3 Specify the properties of the type code, and then press Enter.

TURNOVER® Lifecycle Manager adds the type code, and then displays it on the Work with Type Codes panel.

### To change a type code for a CM scheme

- 1 Select the type code with option 2, and then press Enter.
- 2 Change the type code information, and then press Enter.

For more information about creating and using CM schemes, see *'Managing object types, schemes and data' on page 86*.

# 21 Managing ILE programs

This section describes how to create and promote Integrated Language Environment (ILE) object types in TURNOVER® Lifecycle Manager. It uses ILE RPG for most of its examples, but the same principles apply to all of the ILE languages.

The recommended record length of the ILE RPG source file is 112 bytes.

## Creating modules

To create and maintain ILE \*MODULE object types, you must use the appropriate type code, shipped with TURNOVER® Lifecycle Manager, for handling these object types. Generally, use this method when a module object needs to be “bound” into several programs or service programs. You are recommended to create a separate library in your application definition for storing your \*MODULE objects, but you can also use your existing object libraries.

A typical TURNOVER® Lifecycle Manager type code entry for a \*MODULE object follows:

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                             |                                      |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|--------------------------------------|
| MM/DD/YY<br>11:22:12                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Change a Type Code                                                                                                          | Your Company, Inc.<br>YOURSYS        |
| Type code . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | <u>RPGMOD</u>                                                                                                               | CM scheme . . . . . <u>*TURNOVER</u> |
| Description . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | <u>ILE RPG Module</u>                                                                                                       | Attribute . . . . . <u>RPGLE</u>     |
| Object type . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | <u>*MODULE</u>                                                                                                              | Source file name . . <u>QRPGLSRC</u> |
| Sequence . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | <u>475</u>                                                                                                                  | Data object . . . . . <u>N</u>       |
| Create command . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | CRTRPGMOD MODULE("&LI"/"&OB") SRCFILE("&SL"/"&SF") SRCMBR("&SM") DBGVIEW("&U1") OPTIMIZE("&U2") TGTRLS("&TR") OPTION("&U3") |                                      |
| <hr/>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                             |                                      |
| <div><div>"&amp;RF" = Reference<br/>"&amp;OB" = Object name<br/>"&amp;LI" = Library name<br/>"&amp;TY" = Type code<br/>"&amp;SM" = Source member name<br/>"&amp;SF" = Source file name<br/>"&amp;SL" = Source library name<br/>"&amp;U0" = Generation options<br/>"&amp;U2" = Optimization level<br/>"&amp;U4" = Source listing options<br/>"&amp;U6" = Unassigned<br/>"&amp;U8" = Binding directory</div><div>"&amp;X0"-"&amp;X9" = Exit<br/>"&amp;TO" = Test object name<br/>"&amp;TL" = Test object library name<br/>"&amp;TR" = Target release<br/>"&amp;FM" = From source member name<br/>"&amp;FF" = From source file name<br/>"&amp;FL" = From source library name<br/>"&amp;U1" = Debugging views<br/>"&amp;U3" = Compiler options<br/>"&amp;U5" = Unassigned<br/>"&amp;U7" = Service program source file<br/>"&amp;U9" = Module source file</div></div> |                                                                                                                             |                                      |
| F3=Exit F4=Prompt/select F7=Applications F12=Cancel F22=Long command                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                             |                                      |

This type code uses the standard IBM-supplied creation command for creating \*MODULE objects. For CBL, CL, or C, change the **Create command**, **Attribute**, and **Source file name** fields.

You can check out and promote \*MODULE object types as you would any other standard IBM i type (such as “non-ILE” RPG and CL programs).

TURNOVER® Lifecycle Manager's cross-reference database supports \*MODULE objects. If you are changing a module object, you can view all of the programs and service programs that refer to that module, so that they can be "re-bound" after the module is recreated.

---

## Creating programs

To create an ILE program object, do one of the following:

- 1 Use the `CRTPGM` command and specify the modules that make up the program.

You must use this method if your program consists of more than one module.

- 2 Use the `CRTBNDxxx` command (where `xxx` is RPG, CBL, CL, or C).

This command creates a program object directly from the program's source, bypassing the module creation process. You can use this command only when your program consists of a single module. This method maintains familiarity with the "old" way of creating programs in one step, as you did before ILE was available.

TURNOVER® Lifecycle Manager supports both methods, even in the same application.

---

## Using CRTPGM to create programs

Using the `CRTPGM` command, you have to specify all of the modules and service programs that make up a program each time you want to recreate it, even if you have updated only a single module.

To address this issue, the TURNOVER® Lifecycle Manager `TCRTPGM` command is as a "wrapper" for the `CRTPGM` command. It uses operating system APIs to specify all of the modules and service programs that comprise your program. The only time you have to specify which modules to include, when you create an ILE program, is when you are adding or removing modules from the list.

The TURNOVER® Lifecycle Manager type code entry that you use for creating ILE RPG programs with this method looks as follows:

| MM/DD/YY                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                     | Change a Type Code                              |                  | Your Company, Inc. |           |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|-------------------------------------------------|------------------|--------------------|-----------|-------------------|--------------------|---------------------|--------------------------|----------------------|----------------------------------|-------------------|------------------------|----------------------------|---------------------------------|--------------------------|-------------------------------|-----------------------------|----------------------------------|----------------------------|-------------------------|----------------------------|--------------------------|--------------------------------|--------------------|--------------------|-------------------------------------|---------------------------|----------------------------|
| 11:32:25                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                     |                                                 |                  | YOURSYS            |           |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |
| Type code                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | . . . . .                           | RPGPGM                                          | CM scheme        | . . . . .          | *TURNOVER |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |
| Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | . . . . .                           | ILE RPG Program                                 | Attribute        | . . . . .          | RPGLE     |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |
| Object type                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | . . . . .                           | *PGM                                            | Source file name | . . . . .          |           |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |
| Sequence                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | . . . . .                           | 500                                             | Data object      | . . . . .          | N         |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |
| Create command                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | . . . . .                           | SOFTTURN/TCRTPGM PGM("&LI"/"&OB") TGTRLS("&TR") |                  |                    |           |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |
| <table border="0"> <tr> <td>"&amp;RF" = Reference</td> <td>"&amp;X0"-"&amp;X9" = Exit</td> </tr> <tr> <td>"&amp;OB" = Object name</td> <td>"&amp;TO" = Test object name</td> </tr> <tr> <td>"&amp;LI" = Library name</td> <td>"&amp;TL" = Test object library name</td> </tr> <tr> <td>"&amp;TY" = Type code</td> <td>"&amp;TR" = Target release</td> </tr> <tr> <td>"&amp;SM" = Source member name</td> <td>"&amp;FM" = From source member name</td> </tr> <tr> <td>"&amp;SF" = Source file name</td> <td>"&amp;FF" = From source file name</td> </tr> <tr> <td>"&amp;SL" = Source library name</td> <td>"&amp;FL" = From source library name</td> </tr> <tr> <td>"&amp;U0" = Generation options</td> <td>"&amp;U1" = Debugging views</td> </tr> <tr> <td>"&amp;U2" = Optimization level</td> <td>"&amp;U3" = Compiler options</td> </tr> <tr> <td>"&amp;U4" = Source listing options</td> <td>"&amp;U5" = Unassigned</td> </tr> <tr> <td>"&amp;U6" = Unassigned</td> <td>"&amp;U7" = Service program source file</td> </tr> <tr> <td>"&amp;U8" = Binding directory</td> <td>"&amp;U9" = Module source file</td> </tr> </table> |                                     |                                                 |                  |                    |           | "&RF" = Reference | "&X0"-"&X9" = Exit | "&OB" = Object name | "&TO" = Test object name | "&LI" = Library name | "&TL" = Test object library name | "&TY" = Type code | "&TR" = Target release | "&SM" = Source member name | "&FM" = From source member name | "&SF" = Source file name | "&FF" = From source file name | "&SL" = Source library name | "&FL" = From source library name | "&U0" = Generation options | "&U1" = Debugging views | "&U2" = Optimization level | "&U3" = Compiler options | "&U4" = Source listing options | "&U5" = Unassigned | "&U6" = Unassigned | "&U7" = Service program source file | "&U8" = Binding directory | "&U9" = Module source file |
| "&RF" = Reference                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | "&X0"-"&X9" = Exit                  |                                                 |                  |                    |           |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |
| "&OB" = Object name                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | "&TO" = Test object name            |                                                 |                  |                    |           |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |
| "&LI" = Library name                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | "&TL" = Test object library name    |                                                 |                  |                    |           |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |
| "&TY" = Type code                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | "&TR" = Target release              |                                                 |                  |                    |           |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |
| "&SM" = Source member name                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | "&FM" = From source member name     |                                                 |                  |                    |           |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |
| "&SF" = Source file name                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | "&FF" = From source file name       |                                                 |                  |                    |           |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |
| "&SL" = Source library name                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | "&FL" = From source library name    |                                                 |                  |                    |           |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |
| "&U0" = Generation options                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | "&U1" = Debugging views             |                                                 |                  |                    |           |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |
| "&U2" = Optimization level                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | "&U3" = Compiler options            |                                                 |                  |                    |           |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |
| "&U4" = Source listing options                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | "&U5" = Unassigned                  |                                                 |                  |                    |           |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |
| "&U6" = Unassigned                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | "&U7" = Service program source file |                                                 |                  |                    |           |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |
| "&U8" = Binding directory                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | "&U9" = Module source file          |                                                 |                  |                    |           |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |
| F3=Exit F4=Prompt/select F7=Applications F12=Cancel F22=Long command                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                     |                                                 |                  |                    |           |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |

Because ILE programs are language-independent, you could just have a single type code that you use to create all ILE program objects using this method. However, for clarity, TURNOVER® Lifecycle Manager provides a type code for each language.

TCRTPGM has the same parameters and values as CRTPGM. It determines the proper module and service program list, then builds and executes the appropriate CRTPGM command string. You can override some of the other command parameters, and your overrides are merged into the command string. If you do not specify a value for a parameter, the CRTPGM command default value for that parameter is used.

When determining the module and service program list to use, TURNOVER® Lifecycle Manager applies the following rules:

- If you are using the TCRTPGM command to create an object you are testing (as when you are working with PWM or as a PDM user option), the command uses the library list to find the version of the program it uses for retrieving the reference information.
- If you are using the TCRTPGM command on a form to promote an object, the command uses, as the reference object, the program you are promoting in the *from* library. If the program is not in that library, the command looks for the program in the target library. If it is not there either, the command looks for the program in the library list, and uses that.

Usually, as long as you want to recreate programs as they exist, with the same modules bound together, you can execute the command with its defaults. If you want to add or remove a module from a program, you must recreate it *in your development library*, from a command line using the IBM i CRTPGM command, and specify the modules you want to bind together. To determine what modules to use, the TURNOVER® Lifecycle Manager form promotion job looks in your *from* library for the reference program. You do not need to specify this information again when you promote the program.

You can check out and promote programs as you would any other object-only type. However, because there is no source, but there is a create command to execute, the default method is always COPR (you can also use MO or CD).

## Using CRTBNDxxx to create programs

Bound programs created using the CRTBNDxxx command do not contain any information, in the object's description, about the source from which they were created.

### Defining the underlying create command

You do not have to use the TCBP command in your create commands. You can specify the IBM i create command (CRTBNDRPG, CRTBNDCBL, and so on) as your default create command. An example of the RPGLE global type code is shown below:

|                                                                           |                                                                                                                          |                                 |
|---------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|---------------------------------|
| MM/DD/YY<br>11:25:32                                                      | <b>Change a Type Code</b>                                                                                                | Your Company, Inc.<br>YOURSYS   |
| Type code . . . . .                                                       | RPGLE                                                                                                                    | CM scheme . . . . . *TURNOVER   |
| Description . . . . .                                                     | ILE RPG Program                                                                                                          | Attribute . . . . . RPGLE       |
| Object type . . . . .                                                     | *PGM                                                                                                                     | Source file name . . . QRPGLSRC |
| Sequence . . . . .                                                        | 500                                                                                                                      | Data object . . . . . N         |
| Create command . . .                                                      | CRTBNDRPG PGM("&LI"/"&OB") SRCFILE("&SL"/"&SF") SRCMBR("&SM") DBGVIEW("&X1") OPTIMIZE("&X0") TGTRLS("&TR") OPTION("&U3") |                                 |
| <hr/>                                                                     |                                                                                                                          |                                 |
| "&RF" = Reference                      "&X0"-"&X9" = Exit                 |                                                                                                                          |                                 |
| "&OB" = Object name                    "&TO" = Test object name           |                                                                                                                          |                                 |
| "&LI" = Library name                   "&TL" = Test object library name   |                                                                                                                          |                                 |
| "&TY" = Type code                      "&TR" = Target release             |                                                                                                                          |                                 |
| "&SM" = Source member name           "&FM" = From source member name      |                                                                                                                          |                                 |
| "&SF" = Source file name              "&FF" = From source file name       |                                                                                                                          |                                 |
| "&SL" = Source library name           "&FL" = From source library name    |                                                                                                                          |                                 |
| "&U0" = Generation options            "&U1" = Debugging views             |                                                                                                                          |                                 |
| "&U2" = Optimization level            "&U3" = Compiler options            |                                                                                                                          |                                 |
| "&U4" = Source listing options        "&U5" = Unassigned                  |                                                                                                                          |                                 |
| "&U6" = Unassigned                    "&U7" = Service program source file |                                                                                                                          |                                 |
| "&U8" = Binding directory             "&U9" = Module source file          |                                                                                                                          |                                 |
| F3=Exit F4=Prompt/select F7=Applications F12=Cancel F22=Long command      |                                                                                                                          |                                 |

You can also use the Post-Form Exit (exit 1) to update source information in the object description.

## Creating service programs

Service programs are handled in the same way as the CRTPGM method for creating programs. The TURNOVER® Lifecycle Manager TCRTSRVPGM command operates in the same way as TCRTPGM.

A typical service program TURNOVER® Lifecycle Manager type code entry looks like this:



| MM/DD/YY                                                             |           | Change a Type Code                                    |                  | Your Company, Inc. |           |
|----------------------------------------------------------------------|-----------|-------------------------------------------------------|------------------|--------------------|-----------|
| 11:03:26                                                             |           |                                                       |                  | YOURSYS            |           |
| Type code                                                            | . . . . . | RPGSRV                                                | CM scheme        | . . . . .          | *TURNOVER |
| Description                                                          | . . . . . | ILE RPG Service Program                               | Attribute        | . . . . .          | RPGLE     |
| Object type                                                          | . . . . . | *SRVPGM                                               | Source file name | . . . . .          |           |
| Sequence                                                             | . . . . . | 490                                                   | Data object      | . . . . .          | N         |
| Create command                                                       | . . . . . | SOFTTURN/TCRTSRVPGM SRVPGM("&LI"/"&OB") TGTRLS("&TR") |                  |                    |           |
|                                                                      |           |                                                       |                  |                    |           |
| &RF = Reference                                                      |           | &X0 "-"&X9 = Exit                                     |                  |                    |           |
| &OB = Object name                                                    |           | &TO = Test object name                                |                  |                    |           |
| &LI = Library name                                                   |           | &TL = Test object library name                        |                  |                    |           |
| &TY = Type code                                                      |           | &TR = Target release                                  |                  |                    |           |
| &SM = Source member name                                             |           | &FM = From source member name                         |                  |                    |           |
| &SF = Source file name                                               |           | &FF = From source file name                           |                  |                    |           |
| &SL = Source library name                                            |           | &FL = From source library name                        |                  |                    |           |
| &U0 = Generation options                                             |           | &U1 = Debugging views                                 |                  |                    |           |
| &U2 = Optimization level                                             |           | &U3 = Compiler options                                |                  |                    |           |
| &U4 = Source listing options                                         |           | &U5 = Unassigned                                      |                  |                    |           |
| &U6 = Unassigned                                                     |           | &U7 = Service program source file                     |                  |                    |           |
| &U8 = Binding directory                                              |           | &U9 = Module source file                              |                  |                    |           |
| F3=Exit F4=Prompt/select F7=Applications F12=Cancel F22=Long command |           |                                                       |                  |                    |           |

ILE service programs are language independent. You could use one type code for all languages. The important thing to consider when setting up a service program type code is the sequence number. Because service programs are created from modules, they should have a higher sequence number than modules. However, because they are used to create programs, they must have a lower sequence number than programs. This ensures that the promotion order on a TURNOVER® Lifecycle Manager form is: modules, service programs, and then programs.

You can check out and promote service programs as you would any other object-only type. However, because there is no source, but there is a create command to execute, the default method is always COPR (of course, you can also use MO or CD). The same is true for programs using the CRTPGM method.

---

## Other ILE object types

### Binding language source (BND)

This is a source-only type that contains statements that you can use when creating service programs. If you want to promote the binding language source separately as a source-only type in TURNOVER® Lifecycle Manager, you must have a TURNOVER® Lifecycle Manager type code entry defined like this:

| MM/DD/YY                                                             | Change a Type Code                  | Your Company, Inc.                   |
|----------------------------------------------------------------------|-------------------------------------|--------------------------------------|
| 11:05:26                                                             |                                     | YOURSYS                              |
| Type code . . . . .                                                  | <u>BND</u>                          | CM scheme . . . . . <u>*TURNOVER</u> |
| Description . . . . .                                                | <u>ILE Binder Language</u>          | Attribute . . . . . <u>BND</u>       |
| Object type . . . . .                                                | <u>*SRC</u>                         | Source file name . . <u>QSRVSRC</u>  |
| Sequence . . . . .                                                   | <u>100</u>                          | Data object . . . . . <u>N</u>       |
| Create command . . .                                                 |                                     |                                      |
| <hr/>                                                                |                                     |                                      |
| "&RF" = Reference                                                    | "&X0"-"&X9" = Exit                  |                                      |
| "&OB" = Object name                                                  | "&TO" = Test object name            |                                      |
| "&LI" = Library name                                                 | "&TL" = Test object library name    |                                      |
| "&TY" = Type code                                                    | "&TR" = Target release              |                                      |
| "&SM" = Source member name                                           | "&FM" = From source member name     |                                      |
| "&SF" = Source file name                                             | "&FF" = From source file name       |                                      |
| "&SL" = Source library name                                          | "&FL" = From source library name    |                                      |
| "&U0" = Generation options                                           | "&U1" = Debugging views             |                                      |
| "&U2" = Optimization level                                           | "&U3" = Compiler options            |                                      |
| "&U4" = Source listing options                                       | "&U5" = Unassigned                  |                                      |
| "&U6" = Unassigned                                                   | "&U7" = Service program source file |                                      |
| "&U8" = Binding directory                                            | "&U9" = Module source file          |                                      |
| F3=Exit F4=Prompt/select F7=Applications F12=Cancel F22=Long command |                                     |                                      |

## Bind directory (BNDDIR)

This is an object-only type; that is, it is moved or duplicated from test to target libraries. You can work with the directory and add directory entries using the CRTBNDDIR and WRKBNDDIR commands. Once a directory is created, the CRTPGM command can reference it to indicate what modules or service programs should be included in the ILE program. To set up a BNDDIR type, you must have a TURNOVER® Lifecycle Manager type entry like this:

| MM/DD/YY                                                             | Change a Type Code                  | Your Company, Inc.                   |
|----------------------------------------------------------------------|-------------------------------------|--------------------------------------|
| 11:50:10                                                             |                                     | YOURSYS                              |
| Type code . . . . .                                                  | <u>BNDDIR</u>                       | CM scheme . . . . . <u>*TURNOVER</u> |
| Description . . . . .                                                | <u>ILE Binding Directory</u>        | Attribute . . . . .                  |
| Object type . . . . .                                                | <u>*BNDDIR</u>                      | Source file name . .                 |
| Sequence . . . . .                                                   | <u>480</u>                          | Data object . . . . . <u>N</u>       |
| Create command . . .                                                 |                                     |                                      |
| <hr/>                                                                |                                     |                                      |
| "&RF" = Reference                                                    | "&X0"-"&X9" = Exit                  |                                      |
| "&OB" = Object name                                                  | "&TO" = Test object name            |                                      |
| "&LI" = Library name                                                 | "&TL" = Test object library name    |                                      |
| "&TY" = Type code                                                    | "&TR" = Target release              |                                      |
| "&SM" = Source member name                                           | "&FM" = From source member name     |                                      |
| "&SF" = Source file name                                             | "&FF" = From source file name       |                                      |
| "&SL" = Source library name                                          | "&FL" = From source library name    |                                      |
| "&U0" = Generation options                                           | "&U1" = Debugging views             |                                      |
| "&U2" = Optimization level                                           | "&U3" = Compiler options            |                                      |
| "&U4" = Source listing options                                       | "&U5" = Unassigned                  |                                      |
| "&U6" = Unassigned                                                   | "&U7" = Service program source file |                                      |
| "&U8" = Binding directory                                            | "&U9" = Module source file          |                                      |
| F3=Exit F4=Prompt/select F7=Applications F12=Cancel F22=Long command |                                     |                                      |

## Promoting a service program with its binder language

You can define a TURNOVER® Lifecycle Manager type code that specifies a service program's binder language source file. This allows you to change the binder language and promote it, along with the service program. To change binder language for a service program, follow these steps:

- 1 Define a new TURNOVER® Lifecycle Manager global type code called ILESRV.

| MM/DD/YY                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                             | Change a Type Code |           | Your Company, Inc. |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |                                                                      |  |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|--------------------|-----------|--------------------|--|-------------------|--------------------|---------------------|--------------------------|----------------------|----------------------------------|-------------------|------------------------|----------------------------|---------------------------------|--------------------------|-------------------------------|-----------------------------|----------------------------------|----------------------------|-------------------------|----------------------------|--------------------------|--------------------------------|--------------------|--------------------|-------------------------------------|---------------------------|----------------------------|----------------------------------------------------------------------|--|
| 11:21:23                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                             |                    |           | YOURSYS            |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |                                                                      |  |
| Type code                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | ILESRV                                                      | CM scheme          | *TURNOVER |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |                                                                      |  |
| Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | ILE Service Program                                         | Attribute          | BND       |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |                                                                      |  |
| Object type                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | *SRVPGM                                                     | Source file name   | QSRVSRC   |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |                                                                      |  |
| Sequence                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 490                                                         | Data object        | N         |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |                                                                      |  |
| Create command                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | TURNLSLSE/TCRTSRVPGM SRVPGM("&LI"/"&OB") EXPORT(*SRCFILE) S |                    |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |                                                                      |  |
| RCFILE("&SL"/"&SF") SRCMBR("&SM") TGTRLS("&TR")                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                             |                    |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |                                                                      |  |
| <table border="0"> <tr> <td>"&amp;RF" = Reference</td> <td>"&amp;X0"-"&amp;X9" = Exit</td> </tr> <tr> <td>"&amp;OB" = Object name</td> <td>"&amp;TO" = Test object name</td> </tr> <tr> <td>"&amp;LI" = Library name</td> <td>"&amp;TL" = Test object library name</td> </tr> <tr> <td>"&amp;TY" = Type code</td> <td>"&amp;TR" = Target release</td> </tr> <tr> <td>"&amp;SM" = Source member name</td> <td>"&amp;FM" = From source member name</td> </tr> <tr> <td>"&amp;SF" = Source file name</td> <td>"&amp;FF" = From source file name</td> </tr> <tr> <td>"&amp;SL" = Source library name</td> <td>"&amp;FL" = From source library name</td> </tr> <tr> <td>"&amp;U0" = Generation options</td> <td>"&amp;U1" = Debugging views</td> </tr> <tr> <td>"&amp;U2" = Optimization level</td> <td>"&amp;U3" = Compiler options</td> </tr> <tr> <td>"&amp;U4" = Source listing options</td> <td>"&amp;U5" = Unassigned</td> </tr> <tr> <td>"&amp;U6" = Unassigned</td> <td>"&amp;U7" = Service program source file</td> </tr> <tr> <td>"&amp;U8" = Binding directory</td> <td>"&amp;U9" = Module source file</td> </tr> <tr> <td>F3=Exit F4=Prompt/select F7=Applications F12=Cancel F22=Long command</td> <td></td> </tr> </table> |                                                             |                    |           |                    |  | "&RF" = Reference | "&X0"-"&X9" = Exit | "&OB" = Object name | "&TO" = Test object name | "&LI" = Library name | "&TL" = Test object library name | "&TY" = Type code | "&TR" = Target release | "&SM" = Source member name | "&FM" = From source member name | "&SF" = Source file name | "&FF" = From source file name | "&SL" = Source library name | "&FL" = From source library name | "&U0" = Generation options | "&U1" = Debugging views | "&U2" = Optimization level | "&U3" = Compiler options | "&U4" = Source listing options | "&U5" = Unassigned | "&U6" = Unassigned | "&U7" = Service program source file | "&U8" = Binding directory | "&U9" = Module source file | F3=Exit F4=Prompt/select F7=Applications F12=Cancel F22=Long command |  |
| "&RF" = Reference                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | "&X0"-"&X9" = Exit                                          |                    |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |                                                                      |  |
| "&OB" = Object name                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | "&TO" = Test object name                                    |                    |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |                                                                      |  |
| "&LI" = Library name                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | "&TL" = Test object library name                            |                    |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |                                                                      |  |
| "&TY" = Type code                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | "&TR" = Target release                                      |                    |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |                                                                      |  |
| "&SM" = Source member name                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | "&FM" = From source member name                             |                    |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |                                                                      |  |
| "&SF" = Source file name                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | "&FF" = From source file name                               |                    |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |                                                                      |  |
| "&SL" = Source library name                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | "&FL" = From source library name                            |                    |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |                                                                      |  |
| "&U0" = Generation options                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | "&U1" = Debugging views                                     |                    |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |                                                                      |  |
| "&U2" = Optimization level                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | "&U3" = Compiler options                                    |                    |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |                                                                      |  |
| "&U4" = Source listing options                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | "&U5" = Unassigned                                          |                    |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |                                                                      |  |
| "&U6" = Unassigned                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | "&U7" = Service program source file                         |                    |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |                                                                      |  |
| "&U8" = Binding directory                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | "&U9" = Module source file                                  |                    |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |                                                                      |  |
| F3=Exit F4=Prompt/select F7=Applications F12=Cancel F22=Long command                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                             |                    |           |                    |  |                   |                    |                     |                          |                      |                                  |                   |                        |                            |                                 |                          |                               |                             |                                  |                            |                         |                            |                          |                                |                    |                    |                                     |                           |                            |                                                                      |  |

## 2 Include the type code in your application.

| MM/DD/YY                      |         | ILESRV Defaults                    |   | Your Company, Inc.                     |  |
|-------------------------------|---------|------------------------------------|---|----------------------------------------|--|
| 11:22:27                      |         | Application: J02M Rel: Ver: Lev: 1 |   | SYSTEM: YOURSYS                        |  |
| Type choices and press Enter. |         |                                    |   |                                        |  |
|                               |         |                                    |   | Field Attributes<br>R=Restrict, L=Lock |  |
| <b>From</b> Source file       | QSRVSRC | Name                               | _ |                                        |  |
| Library                       | *PGMR   | Name, *PGMR                        | L |                                        |  |
| <b>Target</b> Source file     | QSRVSRC | Name                               | _ |                                        |  |
| Library                       | TESTSRC | Name                               | L |                                        |  |
| <b>From</b> object library    | *PGMR   | Name, *PGMR                        | L |                                        |  |
| <b>Target</b> object library  | TESTOBJ | Name                               | L |                                        |  |
| Naming mask                   | AP##### |                                    | L |                                        |  |
| Delete from object            | _       | Y, N                               | _ |                                        |  |
| Delete from source            | _       | Y, N                               | _ |                                        |  |
| Method                        |         | CSCO, MO, CD, COPR                 | _ |                                        |  |
|                               |         | CSMO, CSCD, CS, *NULL              |   |                                        |  |
| F3=Exit F12=Cancel            |         | More...                            |   |                                        |  |

The name of the binder language source member in the application type code must match the name of the service program that you want to change.

- 3 Change the binder language by adding the service program to the Programmer Worklist using the ILESRV type code. Check out the binder language by selecting the service program with option 21. This copies the binder language to your development library. Select the service program in development using option 32 to open the binder language for editing.
- 4 Make any desired changes to the binder language.
- 5 Use CRTSRVPGM in development on the command line, to recreate the service program, using the updated binder language.
- 6 To promote the new service program, together with its binder language, add it to a TURNOVER® Lifecycle Manager form by selecting it with option 46.

When you run the form, both the binder language and the bound service program are promoted to the target library, using method CSCO, CSCD, or CSMO as defined by the ILESRV application type code.

## Work flow for managing ILE objects

### Setting up TURNOVER® Lifecycle Manager to manage ILE objects

Before you can begin handling ILE objects with TURNOVER® Lifecycle Manager, there is some setup work you have to perform:

- 1 Make sure that the TURNOVER® Lifecycle Manager application you plan to use contains the appropriate ILE type codes you will need.

This is a list of the ILE type codes that are included with TURNOVER® Lifecycle Manager:

| For COBOL | For CL  | For SQL C   | For SQL COBOL |
|-----------|---------|-------------|---------------|
| CBLLE     | CLLE    | SQLC        | SQLCBI        |
| CBLMOD    | CLPMOD  | SQLCM       | SQLCBM        |
| CBLSRV    | CLPSRV  | SQLCS       | SQLCBS        |
| CBLPGM    | CLPPGM  |             |               |
| For C     | For RPG | For SQL RPG | For Binding   |
| CLE       | RPGLE   | SQLRPI      | BND           |
| CLEMOD    | RPGMOD  | SQLRPM      | BNDDIR        |
| CLESRV    | RPGSRV  | SQLRPS      |               |
| CLEPGM    | RPGPGM  |             |               |

- 2 Make sure the source files you use have a record length of 112.

Many companies use a source file (for example, QRPGLSRC for RPG ILE source) that they create with the suggested record length of 112. Specify the name of this source file in the appropriate type codes for ILE.

- 3 Create a Programmer Worklist to use for checking out, editing, and promoting the ILE objects you will manage. Specify the programmer as the resource, the application as the one you have prepared for ILE, and supply a reference number (usually the number of a TURNOVER® Lifecycle Manager project task). If you are using the TURNOVER® Lifecycle Manager project system, you can select a task with option 20 to create a worklist for it.

**Note** You can set **List sequence** for your Programmer Worklist defaults (F18 on the worklist) to \*CRTSEQ. This shows items on your worklist in the order they will be created: modules first, followed by the service programs and programs into which you will bind them. The sequence parameter in the global type codes specifies the correct create order for each object type (module, 475; service program, 490; program, 500).

## Handling individual bound modules

Use this approach for objects that you plan to create such that one source member is created, compiled into a module, and immediately bound into its own program, all in one operation. In this case, only one type code is needed for each executable bound program.

- 1 Add the module names to the worklist as object type RPGLE or CLLE.
- 2 Check out the source for the modules (option 21).
- 3 Edit the source in the development library (option 32).
- 4 Compile the source in development library (option 36).
- 5 Add the module name(s) to the TURNOVER® Lifecycle Manager form for the first application level (option 46).
- 6 Run the form to promote the bound modules and their source members to the first application level (option 47).
- 7 Copy the form to the next level (option 43).
- 8 Run the form to promote to the next level (option 47).
- 9 Copy the form and run the form for the remaining unlocked levels.

## Binding several modules together

Do the following steps for modules that you will link together into programs or service programs.

- 1 Add the module names to the worklist as object type RPGMOD or CLPMOD.
- 2 Add to the worklist the program that will contain the linked modules, using object type RPGPGM or CLPPGM. Or, if modules are to be linked into a service program, add the name of the service program to the list as object type RPGSRV or CLPSRV.
- 3 Check out each module and program or service program (option 21).
- 4 Edit each module's source in development (option 32).
- 5 Compile each module in development (option 36).

Next, create new service programs or programs with the appropriate IBM i command.

- 6 Type CRTPGM or CRTSRVPGM on a TURNOVER® Lifecycle Manager command line, and then press F4 to prompt. (The example shows CRTPGM.)

**Create Program (CRTPGM)**

Type choices, press Enter.

|                              |            |                            |
|------------------------------|------------|----------------------------|
| Program . . . . .            |            | Name                       |
| Library . . . . .            | *CURLIB    | Name, *CURLIB              |
| Module . . . . .             | *PGM       | Name, generic*, *PGM, *ALL |
| Library . . . . .            |            | Name, *LIBL, *CURLIB...    |
| + for more values            |            |                            |
| Text 'description' . . . . . | *ENTMODTXT |                            |

Bottom

F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel  
F13=How to use this display   F24=More keys

- 7 Use the **Module** parameter to provide names of all the modules you are linking. To enter additional modules and libraries, use **+ for more values**.

You must identify all the modules that you are linking, if:

- this is the first time you are linking them, or
- you are adding or removing modules from a program or a previously bound service program.

If a program is to be recreated with the same module list, then you can rebind as usual from the worklist using option 36. TURNOVER® Lifecycle Manager looks at the object found in the library list to determine the correct module list.

- 8 Press Enter to link the modules together into a service program or program in the development object library.
- 9 Use option 46 to add to a TURNOVER® Lifecycle Manager form the names of all modules (RPGMOD, CLPMOD...) and all programs or service programs (RPGPGM, RPGSRV, CLPPGM, CLPSRV) into which the modules will be linked.

TURNOVER® Lifecycle Manager can order the form correctly with modules coming first, followed by service programs, and then programs. Also, TURNOVER® Lifecycle Manager links the modules as you specified when you created their program or service program. During cross-reference checking for the form, TURNOVER® Lifecycle Manager can add, to the form, additional programs or service programs that need to be recreated because their modules have changed.

- 10 Run the TURNOVER® Lifecycle Manager form to promote to the first level (option 47).
- 11 Copy the form to the next level (option 43).
- 12 Run the form to promote to the next level (option 47).
- 13 Copy the form, and then run it for the remaining unlocked levels.

---

## Converting existing objects to ILE objects

To convert existing object to ILE objects, do one of the following:

- You can manually convert existing objects to ILE objects using the TURNOVER® Lifecycle Manager Programmer Worklist Manager (PWM) (see *'Using the worklist for manual conversion' on page 358*). However, even with TURNOVER® Lifecycle Manager and PWM, the manual process requires many steps.
- The Convert Object Type (TCVTOBJTYP) utility automates the conversion process.

---

### Using the TCVTOBJTYP utility for automated conversion

The Convert Object Type (TCVTOBJTYP) utility converts an object from one type to another. This utility is particularly useful for converting RPG to RPGLE, and RPG to RPGMOD.

The conversion requires these steps:

- 1 Create a type code conversion table, with entries that specify the command to be executed to convert the object from one type to another.
- 2 Verify that the type codes exist at all levels of your application definition(s).
- 3 Create user-defined options to run the TCVTOBJTYP utility from your worklist or from within PDM.
- 4 Perform the conversion, using one of the user-defined options you created to run the utility.

### Creating conversion table entries

To manage entries in the type code conversion table, you must have TURNOVER® Lifecycle Manager system defaults authority.

All the usual TURNOVER® Lifecycle Manager rules apply to the TCVTOBJTYP utility; that is, all the type codes necessary for this utility to function properly must exist at the global level and at each level of the application (including permanently locked levels) that you will use to perform the conversion.

- 1 On the TURNOVER® Lifecycle Manager main menu, select option 8 for the Utility menu, and then select option 13 (Work with type code conversion table).

The Work with Type Code Conversion Table panel appears.

```

MM/DD/YY 15:12:52      Work with Type Code Conversion Table      Your Company, Inc.
                                                                SLS

Type choices, press Enter.

2=Change  3=Copy  4=Delete  5=Display

   Convert      Convert      Delete      Check
   From Type    To Type      Type        Out   Copy  Conversion Command
_  RPG          RPGLE      RPGCPY       Y    N    CVTRPGSRC FROMFILE("&PL"/ . . .
_  RPG          RPGMOD      RPGCPY       Y    N    CVTRPGSRC FROMFILE("&PL"/ . . .

                                                                Bottom

F3=Exit  F6=Add  F12=Cancel  F21=System command

```

- 2 Use F6 to create an entry (for example, the entry for converting RPG to RPGLE).

For more information, see “Chapter 10: Utility Menu” in the *TURNOVER® Lifecycle Manager User Guide*.

## Verifying type codes

Ensure that the type codes you specify in your conversion table entry exist at each level of the application (including permanently locked levels) that you will be using to perform the conversion. This includes any type code you have defined as the **Type code to delete** in the conversion table entry.

## Creating user-defined options

To perform the conversion, run the `TCVTOBJTYP` utility by selecting items with user-defined options, either from the Programmer Worklist, or from within PDM.

Create one or more of the user-defined options, depending on which ones you think you will use, specifying the `TCVTOBJTYP` utility as shown here:

| Utility                                 | Use                                                                                                                                                                                 |
|-----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Programmer Worklist user-defined option | <code>TCVTOBJTYP OBJ("&amp;OB") FROMTYPE("&amp;OA")</code><br><code>TOTYPE(*SELECT) REF("&amp;RF") APPL("&amp;A")</code><br><code>REL(-11) VER(-22) LEV(-33) PGMR("&amp;PG")</code> |
| Converting from PDM Work with Objects   | <code>TCVTOBJTYP OBJ(&amp;N) FROMTYPE(&amp;A)</code><br><code>TOTYPE(*SELECT)</code>                                                                                                |
| Converting from PDM Work with Members   | <code>TCVTOBJTYP OBJ(&amp;N) FROMTYPE(&amp;T)</code><br><code>TOTYPE(*SELECT)</code>                                                                                                |



Whether you convert from your Worklist or from within PDM, you must create a user-defined option for the specific environment you choose before proceeding. The following instructions assume you have already created the option as shown above.

## Performing the conversion

To convert objects, you select them either from your Programmer Worklist, or from object or member lists within PDM, using the appropriate user-defined option you created in Step 3.

### Converting using the Programmer Worklist

- 1 Add the object to the Programmer Worklist using the “old” type code (for example, RPG), but do not check it out. Select the production-level object with the Worklist user-defined option you created (instead of using option 21). Press Enter.

The TCVTOBJTYP Type Code Selection panel appears.

- 2 Indicate the type code to which you want the object converted by selecting the desired conversion table entry. Press Enter.

The conversion process takes place interactively. When the conversion is finished, the worklist shows the new object type. The object has been converted to the type code you selected, and it has been checked out (if you specified ☒ for **Checkout** in the conversion table entry).

### Converting using PDM

- 1 In the Programmer Worklist, use F20 to go to PDM. Do not add the item to the worklist.
- 2 Select option 2, Work with Objects or option 3, Work with Members, and then find the object or member you want to convert.
- 3 Select the object or member with the appropriate PDM user-defined option you created. (The option is different depending on whether you are in Work with Objects or Work with Members.) Press Enter.

The TCVTOBJTYP Type Code Selection popup panel appears.

- 4 Indicate the type code to which you want the object converted by selecting the desired conversion table entry. Press Enter.

The conversion process takes place interactively. When the conversion is finished, return to the Programmer Worklist. The object is added as the type code you selected, and it is checked out (if you specified ☒ for **Checkout** in the conversion table entry).

## What happens to the “old” source member?

When you first created an entry in the type code conversion table, you may have entered a type code you wanted to delete. If you did, when a form is created for the highest unlocked level (Production) of the application, a D delete line is added to the form for the object being converted. The **Type code to delete** is used to remove the old source member (for example, the RPG member that was converted). In most cases, you will have defined RPGCPY as the type code you want deleted, but your company may use some other RPXXXX type code to designate a source-only type for RPG members.

If you did not specify a **Type code to delete**, then a D delete line is not added to the highest unlocked level (Production) form.

### Recommendation

Enter a value for the Type code to delete parameter in the conversion table entry. As long as the highest unlocked level of the application being used is set to archive source on line, TURNOVER® Lifecycle Manager archives the “old” source member.

---

## Using the worklist for manual conversion

You can also use the worklist to manage the activities involved in converting existing objects to ILE objects.

- 1 Add the name of the objects you want to convert to your worklist as object type RPGLE.
- 2 Check out the source (option 21).

There is no need to copy source to development during checkout.

- 3 Convert the source members from the RPG in production library directly into your development library, using the CVTRPGSRC command (or some other conversion tool) and supplying the appropriate library names.
- 4 Edit the converted source members in your development library and make any necessary changes (option 32).
- 5 Compile the source members into a bound module in development (option 36).
- 6 Add the module names to a TURNOVER® Lifecycle Manager form for the first application level (option 46).
- 7 Run the form to promote the source and bound modules to the first level (option 47).
- 8 Copy and run the form for any additional unlocked levels.

### To delete the original RPG source

If you want to delete the original RPG source after the conversion, do the following:

- 1 Add the RPG source names to the worklist as a *source-only* type (that is, type code RPGCPY). Do not check it out.

- 2 Add the source names you are deleting to a TURNOVER® Lifecycle Manager form (option 46 beside the *source-only* type at the level where you want to delete the source).
- 3 Edit the form (option 42).
- 4 Select with option 2 the *source-only* lines for RPG source you are deleting.
- 5 For each line, change the **Code** to D for delete. (F7 lets you update the top portion.) Set **Delete test object** and **Delete test source** to N.
- 6 Run the TURNOVER® Lifecycle Manager form to delete the RPG source members at the target level (option 47).
- 7 If you want to delete source at any other levels of your application, create and run forms for those levels according to these instructions. If you are archiving source in your application, TURNOVER® Lifecycle Manager archives the old source before deleting it.

## 22 Menus, DFUs, and queries

This section describes how to use TURNOVER® Lifecycle Manager with some special object types.

Each of these types is somewhat different from normal IBM i objects, and requires special treatment as described in this section. You might find this information helpful if you are running System/36 environment applications on the IBM i.

In TURNOVER® Lifecycle Manager, you can create special TURNOVER® Lifecycle Manager types (see “Managing type code definitions” in the *TURNOVER® Lifecycle Manager User Guide*). If you have been working with TURNOVER® Lifecycle Manager for some time, you might need to update the type entries.

---

### IBM i menus

An IBM i menu is a composite object comprising:

- a menu object
- a display file
- a message file
- a source file containing menu commands, used in the process of creating menus.

The type entries that represent these are `MENU`, `MNUDDS`, `MNUMSG`, and `MNUCMD`. These type entries should be set up as illustrated on the panels.

## Menu object (MENU)

| MM/DD/YY                                                                                                                                                                                                                                                                                                                                                                  | Change a Type Code                                  | Your Company, Inc. |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|--------------------|
| 16:46:58                                                                                                                                                                                                                                                                                                                                                                  |                                                     | SLS                |
| Type code                                                                                                                                                                                                                                                                                                                                                                 | <u>MENU</u>                                         |                    |
| Description                                                                                                                                                                                                                                                                                                                                                               | <u>Menu</u>                                         |                    |
| Object type                                                                                                                                                                                                                                                                                                                                                               | *MENU Attribute....                                 |                    |
| Sequence                                                                                                                                                                                                                                                                                                                                                                  | <u>550</u>                                          |                    |
| Data object                                                                                                                                                                                                                                                                                                                                                               | <u>N</u>                                            |                    |
| Source file name                                                                                                                                                                                                                                                                                                                                                          |                                                     |                    |
| Create command                                                                                                                                                                                                                                                                                                                                                            | <u>CRTMNU MENU("&amp;LI"/"&amp;OB") TYPE(*DSPF)</u> |                    |
| <u>DSPF(*LIBL/*MENU) MSGF(*LIBL/*MENU)</u>                                                                                                                                                                                                                                                                                                                                |                                                     |                    |
| "&OB"=Object name                      "&TO"=Test object name<br>"&LI"=Library name                    "&TL"=Test object library name<br>"&TY"=Type code<br>"&SM"=Source member name            "&FM"=From source member name<br>"&SF"=Source file name                "&FF"=From source file name<br>"&SL"=Source library name            "&FL"=From source library name |                                                     |                    |
| F3=Exit    F4=Prompt command    F12=Cancel                                                                                                                                                                                                                                                                                                                                |                                                     |                    |

## Menu commands (MNUCMD)

| MM/DD/YY                                                                                                                                                                                                                                                                                                                                                                  | Change a Type Code         | Your Company, Inc. |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|--------------------|
| 16:46:58                                                                                                                                                                                                                                                                                                                                                                  |                            | SLS                |
| Type code                                                                                                                                                                                                                                                                                                                                                                 | <u>MNUCMD</u>              |                    |
| Description                                                                                                                                                                                                                                                                                                                                                               | <u>Menu Command Source</u> |                    |
| Object type                                                                                                                                                                                                                                                                                                                                                               | *SRC Attribute.....        |                    |
| Sequence                                                                                                                                                                                                                                                                                                                                                                  | <u>100</u>                 |                    |
| Data object                                                                                                                                                                                                                                                                                                                                                               | <u>N</u>                   |                    |
| Source file name                                                                                                                                                                                                                                                                                                                                                          |                            |                    |
| Create command                                                                                                                                                                                                                                                                                                                                                            |                            |                    |
| "&OB"=Object name                      "&TO"=Test object name<br>"&LI"=Library name                    "&TL"=Test object library name<br>"&TY"=Type code<br>"&SM"=Source member name            "&FM"=From source member name<br>"&SF"=Source file name                "&FF"=From source file name<br>"&SL"=Source library name            "&FL"=From source library name |                            |                    |
| F3=Exit    F4=Prompt command    F12=Cancel                                                                                                                                                                                                                                                                                                                                |                            |                    |

## Menu message file (MNUMSG)

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                             |                           |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|---------------------------|
| MM/DD/YY<br>16:46:58                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | <b>Change a Type Code</b>   | Your Company, Inc.<br>SLS |
| Type code                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | <u>MNUMSG</u>               |                           |
| Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | <u>Menu Message File</u>    |                           |
| Object type                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | <u>*MSGF</u> Attribute..... |                           |
| Sequence                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | <u>500</u>                  |                           |
| Data object                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | <u>N</u>                    |                           |
| Source file name                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                             |                           |
| Create command                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                             |                           |
| <hr/> <p>             "&amp;OB"=Object name                      "&amp;TO"=Test object name<br/>             "&amp;LI"=Library name                      "&amp;TL"=Test object library name<br/>             "&amp;TY"=Type code<br/>             "&amp;SM"=Source member name              "&amp;FM"=From source member name<br/>             "&amp;SF"=Source file name                  "&amp;FF"=From source file name<br/>             "&amp;SL"=Source library name              "&amp;FL"=From source library name           </p> |                             |                           |
| <hr/> <p>F3=Exit    F4=Prompt command    F12=Cancel</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                             |                           |

## Menu display file (MNUDDS)

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                       |                           |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|---------------------------|
| MM/DD/YY<br>16:46:58                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | <b>Change a Type Code</b>                             | Your Company, Inc.<br>SLS |
| Type code                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | <u>MNUDDS</u>                                         |                           |
| Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | <u>Menu Display File</u>                              |                           |
| Object type                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | <u>*FILE</u> Attribute.....                           |                           |
| Sequence                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | <u>450</u>                                            |                           |
| Data object                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | <u>N</u>                                              |                           |
| Source file name                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                       |                           |
| Create command                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | <u>CRTDSPF FILE("&amp;LI"/"&amp;OB")</u>              |                           |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | <u>SRCFILE("&amp;SL"/"&amp;SF") SRCMBR("&amp;SM")</u> |                           |
| <hr/> <p>             "&amp;OB"=Object name                      "&amp;TO"=Test object name<br/>             "&amp;LI"=Library name                      "&amp;TL"=Test object library name<br/>             "&amp;TY"=Type code<br/>             "&amp;SM"=Source member name              "&amp;FM"=From source member name<br/>             "&amp;SF"=Source file name                  "&amp;FF"=From source file name<br/>             "&amp;SL"=Source library name              "&amp;FL"=From source library name           </p> |                                                       |                           |
| <hr/> <p>F3=Exit    F4=Prompt command    F12=Cancel</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                       |                           |

## Application definition

Make sure that your application definition identifies all of these types at each level (see “Working with Application Definitions” in the *TURNOVER® Lifecycle Manager User Guide*).

## Menu object (MENU)

|                               |                                              |                       |   |
|-------------------------------|----------------------------------------------|-----------------------|---|
| MM/DD/YY                      | <b>MENU Defaults</b>                         | Your Company, Inc.    |   |
| 11:16:55                      | Application: SLSG Rel: Ver: Lev: SYSTEM: SLS |                       |   |
| Type choices and press Enter. |                                              |                       |   |
|                               |                                              | Field Attribute       |   |
|                               |                                              | R=Restrict, L=Locked  |   |
| From Source file              | <u>QMNUSRC</u>                               | Name                  | L |
| Library                       | <u>*PGMR</u>                                 | Name, *PGMR           | L |
| Target Source file            | <u>QMNUSRC</u>                               | Name                  | L |
| Library                       | <u>QGPL</u>                                  | Name, *OBJLIB         | L |
| From object library           | <u>*PGMR</u>                                 | Name, *PGMR           | — |
| Target object library         | <u>QGPL</u>                                  | Name, *PGMR           | — |
| Delete test object            | <u>N</u>                                     | Y, N                  | L |
| Delete test source            | <u>Y</u>                                     | Y, N                  | L |
| Method                        | <u>CSCO</u>                                  | CSCO, MO, CD, COPR    | L |
|                               |                                              | CSMO, CSCD, CS, *NULL |   |
| F3=Exit F12=Cancel            |                                              |                       |   |

## Menu display file (MNUDDS)

|                               |                                              |                       |   |
|-------------------------------|----------------------------------------------|-----------------------|---|
| MM/DD/YY                      | <b>MNUDDS Defaults</b>                       | Your Company, Inc.    |   |
| 11:16:55                      | Application: SLSG Rel: Ver: Lev: SYSTEM: SLS |                       |   |
| Type choices and press Enter. |                                              |                       |   |
|                               |                                              | Field Attribute       |   |
|                               |                                              | R=Restrict, L=Locked  |   |
| From Source file              | <u>QMNUSRC</u>                               | Name                  | L |
| Library                       | <u>*PGMR</u>                                 | Name, *PGMR           | L |
| Target Source file            | <u>QMNUSRC</u>                               | Name                  | L |
| Library                       | <u>QGPL</u>                                  | Name, *OBJLIB         | L |
| From object library           | <u>*PGMR</u>                                 | Name, *PGMR           | — |
| Target object library         | <u>QGPL</u>                                  | Name, *PGMR           | — |
| Delete test object            | <u>N</u>                                     | Y, N                  | L |
| Delete test source            | <u>Y</u>                                     | Y, N                  | L |
| Method                        | <u>CSCO</u>                                  | CSCO, MO, CD, COPR    | L |
|                               |                                              | CSMO, CSCD, CS, *NULL |   |
| F3=Exit F12=Cancel            |                                              |                       |   |

## Menu messages (MNUMSG)

| MM/DD/YY                      | MNUMSG Defaults                              | Your Company, Inc.    |   |
|-------------------------------|----------------------------------------------|-----------------------|---|
| 11:16:55                      | Application: SLSG Rel: Ver: Lev: SYSTEM: SLS |                       |   |
| Type choices and press Enter. |                                              |                       |   |
|                               |                                              | Field Attribute       |   |
|                               |                                              | R=Restrict, L=Locked  |   |
| From Source file              | _____                                        | Name                  | L |
| Library                       | _____                                        | Name, *PGMR           | L |
| Target Source file            | _____                                        | Name                  | L |
| Library                       | _____                                        | Name, *OBJLIB         | L |
| From object library           | *PGMR                                        | Name, *PGMR           | — |
| Target object library         | QGPL                                         | Name, *PGMR           | — |
| Delete test object            | <u>N</u>                                     | Y, N                  | L |
| Delete test source            | <u>Y</u>                                     | Y, N                  | L |
| Method                        | <u>CD</u>                                    | CSCO, MO, CD, COPR    | L |
|                               |                                              | CSMO, CSCD, CS, *NULL |   |
| F3=Exit F12=Cancel            |                                              |                       |   |

## Menu commands (MNUCMD)

| MM/DD/YY                      | MNUCMD Defaults                              | Your Company, Inc.    |   |
|-------------------------------|----------------------------------------------|-----------------------|---|
| 11:16:55                      | Application: SLSG Rel: Ver: Lev: SYSTEM: SLS |                       |   |
| Type choices and press Enter. |                                              |                       |   |
|                               |                                              | Field Attribute       |   |
|                               |                                              | R=Restrict, L=Locked  |   |
| From Source file              | QMUSRC                                       | Name                  | L |
| Library                       | *PGMR                                        | Name, *PGMR           | L |
| Target Source file            | QMUSRC                                       | Name                  | L |
| Library                       | QGPL                                         | Name, *OBJLIB         | L |
| From object library           | *PGMR                                        | Name, *PGMR           | — |
| Target object library         | QGPL                                         | Name, *PGMR           | — |
| Delete test object            | <u>N</u>                                     | Y, N                  | L |
| Delete test source            | <u>Y</u>                                     | Y, N                  | L |
| Method                        | <u>CS</u>                                    | CSCO, MO, CD, COPR    | L |
|                               |                                              | CSMO, CSCD, CS, *NULL |   |
| F3=Exit F12=Cancel            |                                              |                       |   |

## Developing IBM i menus using the worklist

- 1 To add an IBM i menu to your worklist, locate the \*MENU object using PDM's Work with Objects, and then select it with the AW or AI user-defined option.

This adds all four components to your worklist. (You must select the \*MENU component with AI or AW for all four to be added.)

- 2 When you create a new menu, add it to your worklist using F6.
- 3 Specify these parameters:

| Parameter | Value          |
|-----------|----------------|
| Object    | Your menu name |



| Parameter             | Value                                                                                                                                       |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| Object type attribute | MENU [MNUCMD   MNUDDS   MNUMSG]                                                                                                             |
| Application code      | Your application code                                                                                                                       |
| Reference             | Defaults to your project and task code (if you use the TURNOVER® Lifecycle Manager project system); or you can type another reference code. |
| Checkout item         | *YES checks out the object when you add it to the worklist.<br><br>*NO adds the object to the worklist without checking it out.             |

It is checkout that actually reserves the new name if an existing object is not found.

- 4 Repeat the process for the other menu components: MNUCMD, MNUDDS, and MNUMSG.

For example, having done this once for each menu component, then TURNOVER® Lifecycle Manager would reserve these names for menu MENU1:

| Name    |        | Description    |
|---------|--------|----------------|
| MENU1QQ | MNUCMD | Command source |
| MENU1   | MNUDDS | Display file   |
| MENU1   | MNUMSG | Message file   |
| MENU1   | MENU   | Menu           |

- 5 Once these are on your worklist you can check out and promote each one using options 21 and 46, respectively.

## Checking out an IBM i menu type

When you check out a menu, add all menu components to the worklist using F6, and then select all components with option 21 on the programmer worklist.

TURNOVER® Lifecycle Manager assigns your project and task code as a default reference.

The menu components are copied to your development library (the “Source checkout library” for the application). After editing the menu, you can add it to a TURNOVER® Lifecycle Manager form using worklist option 46.

## Adding IBM i menus to a TURNOVER® Lifecycle Manager form

Select each component with worklist option 46 to add them to a form.

If you manually create the TURNOVER® Lifecycle Manager form for the menu object, use the reference code you used when you checked out or reserved the object name.

When the form runs, the menu components are copied to your target library, and then the CRTMNU (create menu) command runs. The MENU type appears last on your form after the MNUCMD, MNUDDS, and MNUMSG.

Some of these types are included on the form only to meet source and object archiving requirements.

## IBM i DFUs

IBM i DFUs are composite object types made up of a program and display file. These can be moved or duplicated into production. In fact, in TURNOVER® Lifecycle Manager, these are *object-only* types (there is no IBM i DFU source) and the only TURNOVER® Lifecycle Manager method possible is MO (Move Object) or CD (Create Duplicate).

TURNOVER® Lifecycle Manager defaults to MO if you specify **Delete test object, Yes**; and it defaults to CD if you specify **Delete test object, No**. You need two type records in your TURNOVER® Lifecycle Manager type file and the application type entries.

### DFUPGM type code

|                                      |                                |                           |
|--------------------------------------|--------------------------------|---------------------------|
| MM/DD/YY<br>16:46:58                 | Change a Type Code             | Your Company, Inc.<br>SLS |
| Type code . . . . DFUPGM             |                                |                           |
| Description . . . . DFU              |                                |                           |
| Object type . . . . *PGM             | Attribute . . . .              |                           |
| Sequence . . . . 900                 |                                |                           |
| Data object . . . . N                |                                |                           |
| Source file name .                   |                                |                           |
| Create command . .                   |                                |                           |
|                                      |                                |                           |
|                                      |                                |                           |
|                                      |                                |                           |
| "&OB"=Object name                    | "&TO"=Test object name         |                           |
| "&LI"=Library name                   | "&TL"=Test object library name |                           |
| "&TY"=Type code                      |                                |                           |
| "&SM"=Source member name             | "&FM"=From source member name  |                           |
| "&SF"=Source file name               | "&FF"=From source file name    |                           |
| "&SL"=Source library name            | "&FL"=From source library name |                           |
|                                      |                                |                           |
| F3=Exit F4=Prompt command F12=Cancel |                                |                           |

## DFUDSP type code

|                      |                           |                           |
|----------------------|---------------------------|---------------------------|
| MM/DD/YY<br>16:46:58 | <b>Change a Type Code</b> | Your Company, Inc.<br>SLS |
|----------------------|---------------------------|---------------------------|

|                            |                  |                     |  |
|----------------------------|------------------|---------------------|--|
| Type code. . . . .         | DFUDSP           |                     |  |
| Description. . . . .       | DFU Display file |                     |  |
| Object type. . . . .       | *FILE            | Attribute . . . . . |  |
| Sequence . . . . .         | 899              |                     |  |
| Data object. . . . .       | N                |                     |  |
| Source file name . . . . . |                  |                     |  |
| Create command . . . . .   |                  |                     |  |

|                           |                                |
|---------------------------|--------------------------------|
| "&OB"=Object name         | "&TO"=Test object name         |
| "&LI"=Library name        | "&TL"=Test object library name |
| "&TY"=Type code           |                                |
| "&SM"=Source member name  | "&FM"=From source member name  |
| "&SF"=Source file name    | "&FF"=From source file name    |
| "&SL"=Source library name | "&FL"=From source library name |

|         |                   |            |
|---------|-------------------|------------|
| F3=Exit | F4=Prompt command | F12=Cancel |
|---------|-------------------|------------|

## IBM i queries

IBM i queries are an object type that can be moved or duplicated into production. In fact, in TURNOVER® Lifecycle Manager, these are “*object-only*” types (there is no IBM i query source) and the only TURNOVER® Lifecycle Manager method possible is MO (Move Object) or CD (Create Duplicate).

TURNOVER® Lifecycle Manager defaults to MO if you say **Delete test object, Yes**; and it defaults to CD if you say **Delete test object, No**.

You just need a single type record in your TURNOVER® Lifecycle Manager type file and the application type entries.

## QRYDFU type code

|                      |                           |                           |
|----------------------|---------------------------|---------------------------|
| MM/DD/YY<br>16:46:58 | <b>Change a Type Code</b> | Your Company, Inc.<br>SLS |
|----------------------|---------------------------|---------------------------|

|                            |                  |                     |  |
|----------------------------|------------------|---------------------|--|
| Type code. . . . .         | QRYDFU           |                     |  |
| Description. . . . .       | Query Definition |                     |  |
| Object type. . . . .       | *QRYDFN          | Attribute . . . . . |  |
| Sequence . . . . .         | 500              |                     |  |
| Data object. . . . .       | N                |                     |  |
| Source file name . . . . . |                  |                     |  |
| Create command . . . . .   |                  |                     |  |

|                           |                                |
|---------------------------|--------------------------------|
| "&OB"=Object name         | "&TO"=Test object name         |
| "&LI"=Library name        | "&TL"=Test object library name |
| "&TY"=Type code           |                                |
| "&SM"=Source member name  | "&FM"=From source member name  |
| "&SF"=Source file name    | "&FF"=From source file name    |
| "&SL"=Source library name | "&FL"=From source library name |

|         |                   |            |
|---------|-------------------|------------|
| F3=Exit | F4=Prompt command | F12=Cancel |
|---------|-------------------|------------|

## 23 Managing SQL objects

TURNOVER<sup>®</sup> Lifecycle Manager can help you manage your database, whether you build it using SQL or DDS.

This section describes how to use TURNOVER<sup>®</sup> Lifecycle Manager to manage these SQL objects:

- tables
- indexes
- views
- triggers
- constraints
- functions
- SQL stored procedures
- externally stored procedures and functions
- *ad hoc* SQL statements
- database security.

One basic concept is common to managing all SQL-related items: You type the SQL Data Definition Language statements (referred to as “DDL”) into a source member, and create the object by executing the `RUNSQLSTM` over the source member. This is similar to typing DDS statements into a source member, and compiling it using the `CRTPF` or `CRTL` commands to create the objects.

While DDS *is* compiled from source, it is better to think of an SQL object as something that is generated from a script. For example, if there is an error in a line of the DDS source, the compiler does not generate that object. With SQL, processing stops, but some source statements may have already been executed, and they cannot be undone. This may leave you with unexpected and undesirable results. For example, the table might get created, but the constraints are not applied.

DDS, by definition, has a one-to-one relationship between the source and the object. In contrast, SQL can generate many objects from a single script.

## About SQL collections

The groups of related tables, indexes and views in a database are called “collections.” (Collections are also sometimes referred to as “schemas.”) A database collection is usually created only once during the application setup process. (Your database collections were probably created before you started using TURNOVER® Lifecycle Manager.)

On an IBM i computer, SQL collections are implemented as libraries. Tables are implemented as physical files in the collection library, and indexes and views as logical files. A database collection also contains the triggers, constraints, functions, and stored procedures, which are implemented as programs and used to manipulate the data in the tables, indexes, and views.

TURNOVER® Lifecycle Manager has no specific support for managing an SQL collection. However, you can have one big script that executes a series of SQL statements to create an entire database, sometimes even populating it with some records, which is sometimes referred to as a “collection.”

TURNOVER® Lifecycle Manager *can* execute that script for you (see [‘Managing programs that require SQL packages’ on page 384](#)). However, when a script is executed in this manner, TURNOVER® Lifecycle Manager has no real knowledge of what the script is going to do. Under such circumstances, TURNOVER® Lifecycle Manager cannot provide the protections and audit trail that most users want.

To use TURNOVER® Lifecycle Manager to manage SQL objects, you might have to rework some of existing scripts, but the benefit is that those objects can then be managed as reliably and effectively.

### Recommendation

To maintain a manageable one-to-one source/object relationship, use one DDL source member to create one SQL object.

## About SQL object names

Names for many types of SQL objects can be much longer than the ten-character IBM i object name limit. SQL objects actually have two names: the *SQL name* and the *system name*.

- The SQL name is used internally by SQL, and the system name is used by non-SQL interfaces, such as high-level language programs and IBM i commands.
- The system name is the name of the IBM i object that appears in your collection library.

When you create an SQL object with a name that is longer than ten characters, the IBM i automatically assigns a corresponding ten-character system name. For example, table `CustSalePart` might be renamed “CustS00001.” The resulting database might be confusing and difficult to manage. Therefore, we make these recommendations about naming your SQL objects:

- Always name the source member the same as the System Name (for example, `CstSalePrt`).
- *For tables or views:* Add a line at the end of the source to tell the system what you want to use as the short system name. For example:

```
Rename Table CustSalePart To System Name CstSalePrt
```

The SQL name *must* be more than 10 characters; otherwise the rename statement will also rename the SQL name. Under these circumstances, you can use the SQL statement `CREATE ALIAS` to create an “alias” for the SQL name.

- *For indexes:*

```
Rename Index CustomerbyTerritory To System Name CustTerr
```

- *For SQL stored procedures, externally stored procedures, and functions:* Use the `Specific` clause. For example:

```
Create Procedure MyLongProcedure Specific MyLongPr
```

- *For constraints:* You do not have to worry about these because the IBM i object isn't visible and the `RMVPFCST` command lets you specify the long SQL Name.

Consider the code for the `CatItemFmt` table below, which shows how you can control not only the SQL and system names, but also the record format name. (This might be helpful in situations where you have RPG or COBOL programs that use native I/O.)

```
Create Table CatItemFmt
( catalogID int not null,
  productID int not null,
  Primary Key( catalogID, productID ) );
Rename Table CatItemFmt to Catalog_Items for System Name CatItems;
```

## About the RUNSQLSTM command's parameter requirements

You can specify additional `RUNSQLSTM` parameters in the type code create command if you need to vary from the command defaults.

### DFTRDBCOL parameter

The `DFTRDBCOL` parameter tells the command where to place your objects and where to look for other unqualified objects being referenced. This is useful because the `RUNSQLSTM` is not really a compiler, and it has no concept of a library list through which it searches for unqualified objects.

### COMMIT parameter

Another parameter to consider is the Commitment Control (`COMMIT`) parameter. The creation of a complex SQL table, for example, may require several statements in the source. The first statement might `CREATE` the table and subsequent statements might apply multiple `CONSTRAINTS`. If the `CREATE` succeeds but one of the `CONSTRAINTS` fails, the `COMMIT` value of `*ALL` will “roll back” the prior successful statement (that is, delete the object that was created).

The `COMMIT` parameter is not activated in the default settings shipped with TURNOVER® Lifecycle Manager. If you choose to activate it, test it thoroughly before implementing it in a live application.

## More information

For more information about executing SQL statements using TURNOVER® Lifecycle Manager, see:

*'Managing externally stored procedures and functions' on page 378*

*'Managing programs that require SQL packages' on page 384.*

---

## Managing tables

An SQL table is implemented on the IBM i computer as a physical file. The most significant difference between a "standard" physical file and a table is the source code used to create it. To create a table, you type a "CREATE TABLE" data definition statement into a source member.

In TURNOVER® Lifecycle Manager, several type codes for managing SQL are shipped already defined for you. The PFTBL type code looks like this:

|                                                             |                                              |                                      |
|-------------------------------------------------------------|----------------------------------------------|--------------------------------------|
| MM/DD/YY<br>10:51:38                                        | <b>Change a Type Code</b>                    | Your Company, Inc.<br>SYSNAME        |
| Type code . . . . .                                         | PFTBL                                        | CM Scheme . . . . . *TURNOVER        |
| Description . . . . .                                       | SQL Table                                    | Attribute . . . . . PFTBL            |
| Object type . . . . .                                       | *FILE                                        | Source file name . . . . . SQLTBLSRC |
| Sequence . . . . .                                          | 410                                          | Data object . . . . . Y              |
| Create command . . . . .                                    | RUNSQLSTM SRCFILE("&SL"/"&SF") SRCMBR("&SM") |                                      |
| COMMIT(*NONE) NAMING(*SYS) DFTRDBCOL("&LI")                 |                                              |                                      |
| <hr/>                                                       |                                              |                                      |
| "&OB" = Object name                                         | "&RF" = Reference                            |                                      |
| "&LI" = Library name                                        | "&TO" = Test object name                     |                                      |
| "&TY" = Type code                                           | "&TL" = Test object library name             |                                      |
| "&SM" = Source member name                                  | "&TR" = Target release                       |                                      |
| "&SF" = Source file name                                    | "&FM" = From source member name              |                                      |
| "&SL" = Source library name                                 | "&FF" = From source file name                |                                      |
| "&U0" = Generation options                                  | "&FL" = From source library name             |                                      |
| "&U2" = ILE Debugging View                                  | "&U1" = Product library (commands)           |                                      |
| "&U4" = PMTFIL Message File                                 | "&U3" = ILE Optimization Level               |                                      |
| "&U6" = Unassigned                                          | "&U5" = Sort Sequence                        |                                      |
| "&U8" = LF Parent, Lib, Members                             | "&U7" = Activation Group                     |                                      |
|                                                             | "&U9" = Unassigned                           |                                      |
| F3=Exit F4=Prompt command F7=Select Applications F12=Cancel |                                              |                                      |

The type code name *must* begin with "PF" or "LF" so that TURNOVER® Lifecycle Manager knows it is a database file (copies data back, handles logical files, and so on).

The default source file name can be modified to suit your internal naming standards.

The table is handled like any other physical file: Data is copied back, logical files (such as Views and Indexes) are recreated, members are put back, triggers and constraints are put back; journaling is stopped and restarted, and so on.

For more information, see *'Managing constraints' on page 373* and *'Managing database security' on page 374.*

---

## Managing indexes and views

Indexes and views are implemented on the IBM i computer as logical files. You manage them the same as you would a table: by typing DDL statements into a source member, and creating them using the RUNSQLSTM command.

TURNOVER® Lifecycle Manager includes the following type codes for managing these objects:

### Type code for an SQL index

|                                                        |                                                                 |                       |                  |
|--------------------------------------------------------|-----------------------------------------------------------------|-----------------------|------------------|
| Type code . . . . .                                    | <u>PFTBL</u>                                                    | CM Scheme . . . . .   | <u>*TURNOVER</u> |
| Description . . . . .                                  | <u>SQL Table</u>                                                | Attribute . . . . .   | <u>PFTBL</u>     |
| Object type . . . . .                                  | <u>*FILE</u>                                                    | Source file name . .  | <u>SQLTBLSRC</u> |
| Sequence . . . . .                                     | <u>410</u>                                                      | Data object . . . . . | <u>Y</u>         |
| Create command . . .                                   | <u>RUNSQLSTM SRCFILE("&amp;SL"/"&amp;SF") SRCMBR("&amp;SM")</u> |                       |                  |
| <u>COMMIT(*NONE) NAMING(*SYS) DFTRDBCOL("&amp;LI")</u> |                                                                 |                       |                  |
|                                                        |                                                                 |                       |                  |

### Type code for an SQL view

|                                                        |                                                                 |                       |                   |
|--------------------------------------------------------|-----------------------------------------------------------------|-----------------------|-------------------|
| Type code . . . . .                                    | <u>LFVIEW</u>                                                   | CM Scheme . . . . .   | <u>*TURNOVER</u>  |
| Description . . . . .                                  | <u>SQL View</u>                                                 | Attribute . . . . .   | <u>LFVIEW</u>     |
| Object type . . . . .                                  | <u>*FILE</u>                                                    | Source file name . .  | <u>SQLVIEWSRC</u> |
| Sequence . . . . .                                     | <u>424</u>                                                      | Data object . . . . . | <u>Y</u>          |
| Create command . . .                                   | <u>RUNSQLSTM SRCFILE("&amp;SL"/"&amp;SF") SRCMBR("&amp;SM")</u> |                       |                   |
| <u>COMMIT(*NONE) NAMING(*SYS) DFTRDBCOL("&amp;LI")</u> |                                                                 |                       |                   |

The name for the type codes can be anything that begins with "LF". The same information about the command parameters and source file names that applies to tables also applies to indexes and views.

TURNOVER® Lifecycle Manager manages indexes and views as it would any logical file. When a table is changed in TURNOVER® Lifecycle Manager, any indexes or views that are built over the table are automatically recreated.

You could use a single type code to manage indexes and views, called something like "LFSQL." However, using two type codes means that the type code clearly distinguishes the type of object, and allows the source and/or objects for each to be stored in different locations.

---

## Managing triggers

SQL triggers can be promoted and maintained by TURNOVER® Lifecycle Manager like any other SQL object. However, triggers should also be stored in the source for the table to which they belong, so that when the table is recreated the triggers are reapplied. TURNOVER® Lifecycle Manager reapplies triggers that are defined with `ADDPFTRG` (such as if you were setting a trigger on a PF), but it does not reapply SQL triggers, because that would require having access to the source. Putting them in the source with the table solves that problem.

The following example shows source for both the table and the trigger:



```
Create Table Tablename
( FIELD1      Dec(7,0) Not Null,
  FIELD2      Dec(7,2) Not Null,
  FIELD3      Char(25) Not Null,
  Primary Key (FIELD1),
  Constraint FIELD2CHK
    Check(FIELD2 >= 0));

Create Trigger Triggername
  AFTER INSERT on Tablename
  (Body of Trigger statement);
```

It is possible to create the table without creating the trigger if there are errors in the source for the trigger. Check the joblog and the file description for details.

---

## Managing constraints

SQL can specify and enforce database rules and integrity. In *'Example: DDL source for a simple database' on page 388*, the `CUSTOMER` table has a constraint that ensures that the value entered into the `CrdLimit` column is always  $\geq 0$ .

SQL also enables referential integrity through use of the “foreign key” clause in SQL. In the `SALE` table below, we have specified that a foreign key constraint exists between the `CustId` column in `SALE` and `CustId` in `CUSTOMER`. If the `CUSTOMER` row is deleted, this ensures that all matching rows in the `SALE` table are also deleted.

TURNOVER® Lifecycle Manager manages both these types of constraints. You can add constraints to a table using either DDL (see *'Object creation sequence' on page 387*) or the IBM i command interface.

### Recommendations

- If you are using DDL source members to create your tables, include in the source for your table the DDL statements necessary for adding the desired constraints. This makes it easy for you to rebuild your database or port it to other platforms.
- If you do not manage your constraints in the same DDL source as your table, you can manipulate constraints by running the `ADDPFCST` and `RMVPFCST` commands as post-run commands on the TURNOVER® Lifecycle Manager form. Because TURNOVER® Lifecycle Manager handles existing constraints automatically, do *not* link the commands to the creation of the corresponding object. They only need to be executed once.
- To change or delete an existing constraint, you must use a post-run command. This also applies regardless of whether the table is built using SQL or DDS. Make the corresponding changes in your DDL, so that the source can always be used to recreate or port the database.
- Add new constraints to the DDL by checking out the source, editing, and promoting back up the line.
- Maintain a full set of all database objects at each level of the application.

For example, when you change only the `SALE` file, TURNOVER® Lifecycle Manager examines the object being replaced and re-establishes the constraints with `CUSTOMER` during the form run. Again, this is another reason for having a full set of all tables, views and indexes in each environment.

The form is also following your data conversion instructions (such as `*MAP *DROP`). If you have the PFTBL in QA, for example, and it has some test data in it, the form will also be testing your data conversion methods. In other words, you are doing a dress rehearsal of the data conversion you will eventually be performing against live data in production; this is another reason for maintaining objects at all levels.

---

## Managing database security

DDL has statements for defining the security of your database objects. The only way to define column-level authority on an IBM i database object is by using DDL statements. By default, TURNOVER® Lifecycle Manager manages your security on objects using reference objects; but, because column-level security can only be applied via DDL, TURNOVER® Lifecycle Manager cannot use a reference object to manage this for you.

Therefore, if you need to apply column-level security, you must assume responsibility for all the object's authority requirements. To do this is, include the necessary DDL statements in the source member *and* to tell TURNOVER® Lifecycle Manager NOT to manage the security by setting the **Authority parameter** in the type code to `S` (use the system defaults). Because you are now managing the security for the object, you must include the DDL statements necessary to define all levels of appropriate authority for the object, including its object-level authorities.

### Recommendations

- Do not use column-level security, and do not manage the security settings in the source. IBM documentation suggests that there is a significant performance hit associated with column-level security. You can often use views with object authority to get the same effect as column-level authority.
- If you must use column-level security, you can automate this in TURNOVER® Lifecycle Manager by creating a type code (for example, PFTBLS), as an copy of the PFTBL type code, and change the type code in the application definition at all levels to force the **Authority parameter** to `S`.

---

## Managing functions and SQL stored procedures

- Managing functions and SQL stored procedures requires that you use a TURNOVER® Lifecycle Manager feature called a *CM scheme*.

- Type codes for these items are *not* shipped with TURNOVER® Lifecycle Manager; you must create them as instructed below. You can use either the `RUNSQLSTM` command, OR a special TURNOVER® Lifecycle Manager version called `TRUNSQLSTM`, for the create command for these type codes. Which one you choose depends on the requirements you have for the creation of the object in the target library. See *'Choosing a create command for functions and SQL stored procedures' on page 376*.

Using SQL, you can create both executable functions and stored procedures. You can write these procedures using either a high-level language such as RPGLE, or by using SQL's Procedural Language (SPL).

When you use the `RUNSQLSTM` command against this source, SQL first translates the source into ILE C with embedded SQL. It is then precompiled using the `CRTSQLCI` command to generate the C code with the embedded SQL. Next, it uses the `CRTCMOD` command to create the ILE module. Finally, it uses the `CRTPGM` (for procedures) or the `CRTSRVPGM` (for functions) to bind the module into a program.

At the same time, an entry is created in the SQL Catalog. This is a group of views over a set of system dictionary files stored in library QSYS2. These views contain additional information about SQL tables, views, indexes, keys, constraints, user-defined types, functions and procedures.

If you try to create a function or an SQL stored procedure that is already defined in this table, you get an error stating that the routine already exists. This means that before every run, you must execute the `DROP PROCEDURE` statement. If the function or SQL stored procedure that you are dropping does not exist, you get another error.

You can create your own change management scheme. For more information, see *'Managing object types, schemes and data' on page 86*.

A CM scheme enables you to control what commands are used for IBM i functions such as `CHKOBJ`, `CMPSRC`, `DLTOBJ`, `GRTOBJAUT`, `MOV OBJ`, and so on. TURNOVER® Lifecycle Manager references these commands as it performs its work.

TURNOVER® Lifecycle Manager includes a special scheme for managing SQL functions and SQL stored procedures that are written using SPL. This scheme checks for the existence of the SQL stored procedure in the catalog and updates it accordingly. This effectively automates the process.

The scheme invokes several programs that help you manage these objects. The `SQLDLTOBJ` and `SQLMOVOBJ` programs run an SQL statement to drop a function or an SQL stored procedure after the program object is moved or deleted. `SQLMOVOBJ` also has special logic to handle form recovery, such that the SQL definition is retrieved and stored in the archive library before the existing object is moved out. If the form goes into recovery, the saved definition (rather than moving the object) is used to recreate the function or SQL stored procedure.

To see what CM schemes are already defined, run the `TWRKCMSCH` command from a TURNOVER® Lifecycle Manager command line (there are no parameters). If you do not see the `SQLCM` scheme, restore it: see *'Restoring the SQLCM scheme' on page 376*.

## Restoring the SQLCM scheme

- 1 Search for a save file named SQLSCH in one of the TURNOVER® Lifecycle Manager libraries (it is shipped in SOFTTURN by default). The programs that support the methods of the scheme are already loaded in SOFTTURN and do not need to be restored.

- 2 From a TURNOVER® Lifecycle Manager command line, run this command:

```
TRSTCMSCH SAVF(SOFTTURN/SQLSCH) TOScheme(*SAVScheme) REPLACE(*NO)
RSTPGMS(*NO) PGMLIB(*Scheme) RSTRSRCLIB(*NO) RSRCLIB(*SAVLIB)
```

- 3 If you have loaded TURNOVER® Lifecycle Manager into a library other than SOFTTURN, you must change where the scheme looks for the programs that run any methods specifying the default program and library names:

```
TCHGCMSCHEM(SQL) DESC('SQL Scheme') DFTPGM(your_lib_name/*TURNOver)
```

- 4 From a TURNOVER® Lifecycle Manager command line, run the TWRKCMSCH command to verify the results.

## Creating the type codes

To see how your SQL function and SQL stored procedure type codes should look, see [‘Example of a type code for creating an SQL function with RUNSQLSTM’ on page 377](#).

Note the following settings:

- You can name the type code anything you like.
- The **CM scheme** points TURNOVER® Lifecycle Manager to the proper scheme.
- The **Object type** for an SQL stored procedure must be \*PGM.
- The **Object Type** for a function must be \*SRVPGM.
- A Y value for **Data object** allows the application definition to “explode” data, which effectively copies the object to multiple libraries during a single promotion.
- A value of 800 for **Sequence** ensures that the form has already processed other SQL items, such as tables and views, before the function or SQL procedure is created.
- The **Create command** can be either the RUNSQLSTM command or the TRUNSQLSTM create command. (The next topic tells you how to choose one of these commands.)

## Choosing a create command for functions and SQL stored procedures

There are times when you need to use the NAMING(\*SYS) and DFTRDBCOL(\*NONE) parameter values in your RUNSQLSTM create command, so that the SQL runtime will use the library list to resolve object references.

However, when you do not specify a default database collection (that is, when DFTRDBCOL is set to \*NONE), TURNOVER® Lifecycle Manager loses control of where SQL builds the objects that your script creates.

To resolve this issue, TURNOVER® Lifecycle Manager uses its own version of the RUNSQLSTM command. It is called TRUNSQLSTM, and it has the same parameters as RUNSQLSTM, along with a few additions (specifically the added CURLIB parameter, which you need in this case. Another parameter called RUNAS is also important for other reasons; see *'Controlling unauthorized database changes' on page 385.*)

When you specify the "&LI" substitution variable as the value for the CURLIB parameter, and combine this with the NAMING (\*SYS) and DFTRDBCOL (\*NONE) parameter values in your type code create command, TURNOVER® Lifecycle Manager resolves the substitution variable from the application definition before submitting the command, allowing SQL to create the objects in a location where object references will resolve successfully.

### Example of a type code for creating an SQL function with RUNSQLSTM

| MM/DD/YY              | Change a Type Code                                         | Your Company, Inc.             |
|-----------------------|------------------------------------------------------------|--------------------------------|
| 11:01:41              |                                                            | SYSNAME                        |
| Type code . . . . .   | SQLFNC                                                     | CM Scheme . . . . . SQL        |
| Description . . . . . | SQL Function                                               | Attribute . . . . . SQL        |
| Object type . . . . . | *SRVPGM                                                    | Source file name . . . QSQLSRC |
| Sequence . . . . .    | 800                                                        | Data object . . . . . Y        |
| Create command . . .  | RUNSQLSTM SRCFILE("&SL"/"&SF") SRCMBR("&SM") COMMIT(*NONE) |                                |
|                       | NAMING(*SQL) DFTRDBCOL("&LI")                              |                                |

### Type code for creating an SQL function with TRUNSQLSTM

| MM/DD/YY              | Change a Type Code                                                                                                | Your Company, Inc.             |
|-----------------------|-------------------------------------------------------------------------------------------------------------------|--------------------------------|
| 11:01:41              |                                                                                                                   | SYSNAME                        |
| Type code . . . . .   | SQLFNC                                                                                                            | CM Scheme . . . . . SQL        |
| Description . . . . . | SQL Function                                                                                                      | Attribute . . . . . SQL        |
| Object type . . . . . | *SRVPGM                                                                                                           | Source file name . . . QSQLSRC |
| Sequence . . . . .    | 800                                                                                                               | Data object . . . . . Y        |
| Create command . . .  | SOFTTURNE/TRUNSQLSTM SRCFILE("&SL"/"&SF") SRCMBR("&SM") COMMIT(*NONE) NAMING(*SYS) DFTRDBCOL(*NONE) CURLIB("&LI") |                                |

### Type code for creating an SQL stored procedure with RUNSQLSTM

| MM/DD/YY              | Change a Type Code                                         | Your Company, Inc.             |
|-----------------------|------------------------------------------------------------|--------------------------------|
| 11:01:41              |                                                            | SYSNAME                        |
| Type code . . . . .   | SQLPRC                                                     | CM Scheme . . . . . SQL        |
| Description . . . . . | SQL Stored Procedure                                       | Attribute . . . . . SQL        |
| Object type . . . . . | *PGM                                                       | Source file name . . . QSQLSRC |
| Sequence . . . . .    | 800                                                        | Data object . . . . . Y        |
| Create command . . .  | RUNSQLSTM SRCFILE("&SL"/"&SF") SRCMBR("&SM") COMMIT(*NONE) |                                |
|                       | NAMING(*SQL) DFTRDBCOL("&LI")                              |                                |

## Type code for creating an SQL stored procedure with TRUNSQLSTM

|                       |                                                                                                                      |                                |
|-----------------------|----------------------------------------------------------------------------------------------------------------------|--------------------------------|
| MM/DD/YY<br>11:01:41  | Change a Type Code                                                                                                   | Your Company, Inc.<br>SYSNAME  |
| Type code . . . . .   | SQLPRC                                                                                                               | CM Scheme . . . . . SQL        |
| Description . . . . . | SQL Stored Procedure                                                                                                 | Attribute . . . . . SQL        |
| Object type . . . . . | *PGM                                                                                                                 | Source file name . . . QSQLSRC |
| Sequence . . . . .    | 800                                                                                                                  | Data object . . . . . Y        |
| Create command . . .  | SOFTTURN/TRUNSQLSTM SRCFILE("&SL"/"&SF") SRCMBR("&SM") CO<br>MMIT(*NONE) NAMING(*SYS) DFTRDBCOL(*NONE) CURLIB("&LI") |                                |

## Managing externally stored procedures and functions

### Important!

- Managing externally stored procedures and functions requires that you use a TURNOVER® Lifecycle Manager feature called a *CM scheme*.
- Type codes for these items are NOT shipped with TURNOVER® Lifecycle Manager; you must create them as instructed below. You can use either the RUNSQLSTM command or a special TURNOVER® Lifecycle Manager version called TRUNSQLSTM, for the create command for these type codes. Which one you choose depends on the requirements you have for the creation of the object in the target library. See [‘Choosing a create command for externally stored procedures and functions’ on page 381](#).

You write an externally stored procedure or function in one of the programming languages on the IBM i computer. You can compile the host language programs to create \*PGM or \*SRVPGM objects. To create an externally stored procedure or function, you must compile the source code for the host language to create a program object.

For TURNOVER® Lifecycle Manager to properly promote your routine’s catalog entries, you *must* do the following:

- Create a source member that uses the CREATE PROCEDURE or CREATE FUNCTION statements, to be run with the RUNSQLSTM or TRUNSQLSTM commands.
- Give your source member the same name as the program that it references.
- Use the SPECIFIC statement to give the routine’s catalog entries the same name as the source member name.

## Type codes for external procedures and functions

TURNOVER® Lifecycle Manager manages externally stored procedures and functions using multiple type codes, as follows:

- One type code to manage the \*PGM or \*SRVPGM object.
- Another type code to manage creation of the catalog entries from a source member.

It is possible to change the logic of an externally stored procedure or function without having to change the catalog entry, as long as the parameters and object attributes do not change. Therefore, you might not have to promote the catalog entries each time you modify the program. However, if you make a change that requires the catalog entry to be re-created, you must check out and modify the `CREATE PROCEDURE` or `CREATE FUNCTION` source as well.

You can use any appropriate type code (such as RPGLE or SQLRPI) to manage the \*PGM object and the current processes that TURNOVER® Lifecycle Manager uses will still be the same. To manage the catalog entries, you must use a TURNOVER® Lifecycle Manager change management scheme. For information about managing catalog entries, see *'Managing catalog entries with a TURNOVER® Lifecycle Manager CM scheme' on page 379*.

---

## Managing catalog entries with a TURNOVER® Lifecycle Manager CM scheme

Using a TURNOVER® Lifecycle Manager change management scheme, you can control what commands are used for IBM i functions such as `CHKOBJ`, `DLTOBJ`, `MOVOBJ`, and so on. TURNOVER® Lifecycle Manager references these commands as it performs its work.

TURNOVER® Lifecycle Manager includes a special scheme just for managing the catalog entries of externally stored procedures and functions. This scheme overrides certain methods to handle the catalog entries:

- The \*`CHKOBJ` method examines the catalog entries for a specified library to let you know if a catalog entry exists for the externally stored procedure or function.
- The \*`DLTOBJ` method issues the `DROP PROCEDURE` or `DROP FUNCTION` statement to remove the catalog entry for an externally stored procedure or function.
- The \*`MOVOBJ` method is used during archiving of existing catalog entries, during distribution, and during recovery processing, as follows:

For archiving and distribution, this method ensures that you have source for the existing catalog entries. It does so by generating SQL DDL source from the existing catalog entries in the archive or distribution library. This provides a safeguard against promoting catalog entries that may have been created without source through IBM i Access Client Solutions.

For archiving, this method also drops the existing catalog entries to allow you to create new entries.

For recovery processing, this method ensures that the catalog entries are recreated properly from archived source, or from source that may have been automatically generated for archive.

- The \*`RTVOBJD` method returns certain information about a catalog entry, such as the date and time that it was created, the description of the entry (from the `COMMENT` statement), and so forth.

- Methods `*CHGOBJ`, `*CHGOBJOWN`, `*CRTDUPOBJ`, and `*GRTOBJAUT` are set to `*NONE` to perform no processing, the reason being that these methods perform tasks that are not relevant to catalog entries.

To see what CM schemes are already defined, run the `TWRKCMSCH` command from a TURNOVER® Lifecycle Manager command line (there are no parameters). If you do not see the `EXTPRC` scheme, see *'Restoring the EXTPRC scheme' on page 380*.

---

## Restoring the EXTPRC scheme

- 1 Search for a save file named `UTEXTPRC` in one of the TURNOVER® Lifecycle Manager libraries (it is shipped in `SOFTTURN` by default).
- 2 From a TURNOVER® Lifecycle Manager command line, run this command:

```
TRSTCMSCH SAVF(SOFTTURN/UTEXTPRC) TOScheme(*SAVSCHEME)
REPLACE(*NO) RSTPGMS(*NO) PGMLIB(*SCHEME) RSTRSRCLIB(*NO)
RSRCLIB(*SAVLIB)
```

- 3 If you have loaded TURNOVER® Lifecycle Manager into a library other than `SOFTTURN`, you must change where the scheme will look for the programs that run any methods specifying the default program and library names:

```
TCHGCMSCHEM SCHEME(EXTPRC) DESC('External Stored Procedure Scheme')
DFTPGM(your_lib_name/*TURNOVER)
```

- 4 Run the `TWRKCMSCH` command to verify the results.

---

## Creating the type codes

To see how your externally stored procedures and functions type code should look, see:

*'Type code for creating an externally stored procedure with RUNSQLSTM' on page 381*

*'Type code for creating an externally stored procedure with TRUNSQLSTM' on page 381*

*'Type code for creating an external function with TRUNSQLSTM' on page 382*

### Notes

- Name the type code for external procedures `EXTPRC`, and the type code for external functions `EXTFUN`, but you can name the type code anything you like. If you plan to use an alternate type code, see *'Service program libraries' on page 382*.
- The **CM scheme** points TURNOVER® Lifecycle Manager to the proper scheme.
- The **Object type** for an externally stored procedure must be `*EXTPRC` and externally stored functions must be `*EXTFUN`.
- A `Y` value for **Data object** allows the application definition to “explode” data, which creates the catalog entries for multiple libraries during a single promotion (optional).



- A value of 800 for **Sequence** ensures that the form has already processed the externally stored procedure or function program, before the catalog entries are created.
- The **Create command** can be either the RUNSQLSTM command or the TRUNSQLSTM create command. See *'Choosing a create command for externally stored procedures and functions' on page 381.*

## Choosing a create command for externally stored procedures and functions

There are times when you need to use the NAMING(\*SYS) and DFTRDBCOL(\*NONE) parameter values in your RUNSQLSTM create command, so that the SQL runtime will use the library list to resolve object references.

However, when you do not specify a default database collection (that is, when DFTRDBCOL is set to \*NONE), TURNOVER® Lifecycle Manager loses control of where SQL builds the objects that your script creates.

To resolve this issue, TURNOVER® Lifecycle Manager uses its own version of the RUNSQLSTM command. It is called TRUNSQLSTM, and it has the same parameters as RUNSQLSTM, along with a few additions (specifically the added CURLIB parameter, which you need in this case. Another parameter called RUNAS is also important for other reasons; see *'Controlling unauthorized database changes' on page 385.*

When you specify the "&LI" substitution variable as the value for the CURLIB parameter, and combine this with the NAMING(\*SYS) and DFTRDBCOL(\*NONE) parameter values in your type code create command, TURNOVER® Lifecycle Manager resolves the substitution variable from the application definition before submitting the command, allowing SQL to create the objects in a location where object references will resolve successfully.

### Type code for creating an externally stored procedure with RUNSQLSTM

|                       |                                                                                             |                                |
|-----------------------|---------------------------------------------------------------------------------------------|--------------------------------|
| MM/DD/YY<br>11:01:41  | <b>Change a Type Code</b>                                                                   | Your Company, Inc.<br>SYSNAME  |
| Type code . . . . .   | EXTPRC                                                                                      | CM Scheme . . . . . EXTPRC     |
| Description . . . . . | External Stored Procedure                                                                   | Attribute . . . . . SQL        |
| Object type . . . . . | *EXTPRC                                                                                     | Source file name . . . QSQLSRC |
| Sequence . . . . .    | 800                                                                                         | Data object . . . . . Y COPR   |
| Create command . . .  | RUNSQLSTM SRCFILE("&SL"/"&SF") SRCMBR("&SM") COMMIT(*NONE)<br>NAMING(*SQL) DFTRDBCOL("&LI") |                                |

### Type code for creating an externally stored procedure with TRUNSQLSTM

|                       |                                                                                                                   |                                |
|-----------------------|-------------------------------------------------------------------------------------------------------------------|--------------------------------|
| MM/DD/YY<br>11:01:41  | <b>Change a Type Code</b>                                                                                         | Your Company, Inc.<br>SYSNAME  |
| Type code . . . . .   | EXTPRC                                                                                                            | CM Scheme . . . . . EXTPRC     |
| Description . . . . . | External Stored Procedure                                                                                         | Attribute . . . . . SQL        |
| Object type . . . . . | *EXTPRC                                                                                                           | Source file name . . . QSQLSRC |
| Sequence . . . . .    | 800                                                                                                               | Data object . . . . . Y COPR   |
| Create command . . .  | SOFTTURNE/TRUNSQLSTM SRCFILE("&SL"/"&SF") SRCMBR("&SM") COMMIT(*NONE) NAMING(*SYS) DFTRDBCOL(*NONE) CURLIB("&LI") |                                |

## Type code for creating an external function with TRUNSQLSTM

|                                                          |                                                                                |                       |                |
|----------------------------------------------------------|--------------------------------------------------------------------------------|-----------------------|----------------|
| Type code . . . . .                                      | <u>EXTFUN</u>                                                                  | CM Scheme . . . . .   | <u>EXTPRC</u>  |
| Description . . . . .                                    | <u>External Function</u>                                                       | Attribute . . . . .   | <u>EXTFUN</u>  |
| Object type . . . . .                                    | <u>*EXTFUN</u>                                                                 | Source file name . .  | <u>QSQLSRC</u> |
| Sequence . . . . .                                       | <u>800</u>                                                                     | Data object . . . . . | <u>Y COPR</u>  |
| Create command . . .                                     | <u>SOFTTURN/TRUNSQLSTM SRCFILE ("&amp;SL"/"&amp;SF") SRCMBR ("&amp;SM") CO</u> |                       |                |
| <u>MMIT (*NONE) DFTRDBCOL (*NONE) CURLIB ("&amp;LI")</u> |                                                                                |                       |                |

## Service program libraries

When using a service program (\*SRVPGM) as the external procedure or function, the service program library name must be hard coded. (This is an IBM limitation.) This makes promotion of procedures and functions that use these a bit more difficult as normal TURNOVER® Lifecycle Manager processing does not modify your source code. To handle this situation, the TRUNSQLSTM command has been enhanced to have the additional parameter Replace Library in Source (RPLLIB).

Using the RPLLIB parameter on the TRUNSQLSTM command and the place holder &LIBRARY in your source code causes the TRUNSQLSTM process to copy your source code into a source file in QTEMP, and then the source in QTEMP is modified to replace the &LIBRARY with the actual library name.

For example, your source code may contain this:

```
...
EXTERNAL NAME '&LIBRARY/OURSRVPGM(PROCEDURE01)'
...
```

When the TRUNSQLSTM command is run, the source is copied to QTEMP and any occurrence of ' &LIBRARY ' is replaced with the actual library name before the statements are executed.

Then define the RPLLIB parameter in the global type code definition, using the appropriate substitution variable.

|                                                                                      |                                                                              |                                     |
|--------------------------------------------------------------------------------------|------------------------------------------------------------------------------|-------------------------------------|
| MM/DD/YY                                                                             | Change a Type Code                                                           | Your Company, Inc.                  |
| 11:01:41                                                                             |                                                                              | SYSNAME                             |
| Type code . . . . .                                                                  | <u>EXTPRC</u>                                                                | CM Scheme . . . . . <u>EXTPRC</u>   |
| Description . . . . .                                                                | <u>External Stored Procedure</u>                                             | Attribute . . . . . <u>SQL</u>      |
| Object type . . . . .                                                                | <u>*EXTPRC</u>                                                               | Source file name . . <u>QSQLSRC</u> |
| Sequence . . . . .                                                                   | <u>800</u>                                                                   | Data object . . . . . <u>Y COPR</u> |
| Create command . . .                                                                 | <u>SOFTTURN/TRUNSQLSTM SRCFILE("&amp;SL"/"&amp;SF") SRCMBR("&amp;SM") CO</u> |                                     |
| <u>MMIT(*NONE) NAMING(*SYS) DFTRDBCOL(*NONE) CURLIB("&amp;LI") RPLLIB("&amp;LI")</u> |                                                                              |                                     |

## TURNOVER® Lifecycle Manager cross-reference

When you use TURNOVER® Lifecycle Manager to promote a program that has been defined as an externally stored procedure or function, as shown in this section, TURNOVER® Lifecycle Manager adds records to the TURNOVER® Lifecycle Manager cross-reference table showing the program as being used by the catalog entry. WRKOBJREF, or option 15 from the worklist, will show the catalog entry as a PGM-CALL from the program. You can use a different type code, as long as the global type code is defined with the object type \*EXTPRC or \*EXTFUN.

## Things to remember

While managing externally stored procedures or functions, bear these points in mind:

- Create the global type code as defined in this section. Assign the type code to all the necessary applications and configure it to use method `CSCO` (Copy Source, Compile Object).
- If you distribute, make sure the applications are setup to distribute source for the type codes `EXTPRC` and `EXTFUN`. Also, ensure that the remote machines have the type codes defined globally with the `EXTPRC` scheme, and that the type codes have been assigned to the correct levels of the remote application. The remote application should also use method `CSCO` for promotion.
- Any type code that you create to manage procedures or functions must have the global object type for those type codes set to `*EXTPRC` or `*EXTFUN`.
- Use a source member to define the `CREATE PROCEDURE` or `CREATE FUNCTION` statements, and make use of the `SPECIFIC` statement. The source member name and the `SPECIFIC` name should match. Do not use library names as qualifiers in this source: the `TRUNSQLSTM` command sets up the `*CURLIB` correctly. If you do need to use library names as qualifiers (as is the case with `*SRVPGMs`), make use of the `RPLLIB` parameter on the `TRUNSQLSTM` command and use `&LIBRARY` instead of the actual library name.
- Promotion archives existing source. If the existing catalog entry does not have source, it is generated in the archive library into a file called `QARCHIVSQL`.
- TURNOVER® Lifecycle Manager cross-reference table entries for the `*EXTPRC` and `*EXTFUN` objects come from the `*PGM` object. Promotion of just the `*EXTPRC` or `*EXTFUN` object will not update the cross-reference table entries. You must promote (or run the cross-reference) for the `*PGM` to see the references. TURNOVER® Lifecycle Manager does not provide cross-reference table entries for `*EXTPRC` or `*EXTFUN` objects that use `*SRVPGMs`.

## Managing materialized query tables

Materialized query table (MQTs) provide another method of improving performance of your queries. An MQT is a table that contains the results of a previously run query, along with the query's definition. It provides a mechanism for improving the response time of complex SQL queries. While it can be thought of as a summary table, what sets an MQT apart from a regular summary table is the fact that the SQE optimizer is aware of it and its relationship to the query and base tables that were specified when it was created and populated. This means that the optimizer considers using the MQT in the access plan of subsequent similar queries if it determines that it is appropriate to do so. Because the MQT is already created, populated, joined, aggregated, and sorted, this can result in significant performance improvements for complex queries.

It is important to understand that an MQT is a table (a physical file with an object type of `*FILE` and an attribute of `PF`) that is in a library (schema) in the System i environment.

While their potential performance efficiencies are appealing for DBAs, query developers, and report users, MQTs have several attributes and limitations that must be understood prior to implementation. Probably most important, you must know that MQTs are not automatically maintained. This means that as the base tables used to populate the MQT change, the data in the MQT does not also change.

The functionality of an MQT is similar to the role of an index. Both objects provide a path to the data that the user is normally unaware of. Unlike an index, a user might directly query the MQT just like a table or view.

For more information about materialized query tables see the IBM Redbook *Getting Started with DB2 Web Query for System i* - SG247214, and the IBM white paper, *Creating and using materialized query tables (MQT) in IBM DB2 for i5/OS Version 2*.

### Recommendation

Create a global type code named `LFMQT` (similar to `LFVIEW`). In doing so, it will eliminate the adding of a COPR form line each time the form is edited or run.

---

## Managing programs that require SQL packages

If you use SQL packages, then you will encounter a situation where program objects that you are promoting need to have SQL packages created or re-created.

There are two ways to handle this situation: post-run commands, or an exit program.

### Post-run command method

When a \*PGM on a form needs to have an SQL package created or re-created during the form run, you can link a post-run command to the form line for your program that executes the `CRTSQLPKG` command appropriately. Once you have done this with your program, TURNOVER® Lifecycle Manager thereafter prompts for the correct post-run command when it encounters that object on a subsequent form.

### Exit program method

You can write an exit program to use with one of the TURNOVER® Lifecycle Manager form processing exit points (for example, Exit 20). Your exit program could sift through your form, identify lines eligible for `CRTSQLPKG` processing, and issue the command for each eligible line. This technique is best employed in situations where more automation is desired.

---

## Executing ad hoc SQL statements

There will be times when you need to execute an SQL statement to make a change, but there is no "object movement" associated with this change. For example, you might want to use the `ALTER TABLE` statement to add a new constraint, or change the column headings of a table.

To execute these SQL statements, use the `SQLSTM` type code.

| MM/DD/YY                                    |           | Change a Type Code   |                      | YOUR COMPANY INC. |            |
|---------------------------------------------|-----------|----------------------|----------------------|-------------------|------------|
| 14:07:26                                    |           |                      |                      | YOURSYS           |            |
| Type code                                   | . . . . . | SQLSTT               | CM scheme            | . . . . .         | *TURNOVER  |
| Description                                 | . . . . . | SQLSTM- TRUNSQLSTM   | Attribute            | . . . . .         | SQL        |
| Object type                                 | . . . . . | *SQLSTM              | Source file name     | . . . . .         | QSQLSRC    |
| Sequence                                    | . . . . . | 800                  | Data object          | . . . . .         | Y *DEFAULT |
| Create command                              | . . . . . | SOFTTURNE/TRUNSQLSTM | SRCFILE("&SL"/"&SF") | SRCMBR("&SM")     |            |
| COMMIT(*NONE) DFTRDBCOL("&LI") RUNAS("&U9") |           |                      |                      |                   |            |

Any source member containing such an *ad hoc* SQL statement is, by default, executed as written, at *every* application level and on *every* machine where this form runs. If you do not want this statement to run multiple times on the same machine, or if you only want this statement to run on certain machines, you must manually handle those scenarios by editing the form before it runs to remove that line, or by not running the form or not distributing it to the remote.

Now do the following:

- 1 Use F6 on the worklist to add a new item. Fill in the new item's name and description; the type code will be SQLSTM.
- 2 Check the item out to reserve its name and to allow promotion.
- 3 In your development environment, type the SQL source code into the member.

Be careful about using option 36; it executes the SQL source. (Remember that for SQL objects, this is not a compile; it is actually executing the SQL source using the `RUNSQLSTM` command.)

- 4 Build and run the form.

## Notes

- If your application distributes and you want this SQL source code to execute on the remote computers as well, you must define this type code to distribute the source member and use a promotion method of CSCO on the remote computer. This also means that the SQL licensed programs must be installed on the remote computer.
- The sequence number of "800" means that this statement is one of the last items that is processed by this form. This helps ensure that other items that might be referenced by this *ad hoc* SQL statement, such as tables, are already in place.
- This promotion might be performing any number of a variety of activities by executing the SQL statements in the source file. TURNOVER® Lifecycle Manager has no way of knowing what these statements are doing, so there is no way to "undo" the effects of processing the statements TURNOVER® Lifecycle Manager executes using this technique.

---

## Controlling unauthorized database changes

A TURNOVER® Lifecycle Manager command called `TRUNSQLSTM` allows you to control the user profile under which people can run *ad hoc* SQL statements. This command contains the basic important parameters of the `RUNSQLSTM` command, but adds some additional parameters and logic to it, to help ensure database integrity. To use this command, specify it in a TURNOVER® Lifecycle Manager type code, like this:

|                                                             |                                                                           |                                         |  |  |
|-------------------------------------------------------------|---------------------------------------------------------------------------|-----------------------------------------|--|--|
| MM/DD/YY<br>14:07:26                                        | <b>Change a Type Code</b>                                                 | YOUR COMPANY INC.<br>YOURSYS            |  |  |
| Type code . . . . .                                         | <u>SQLSTT</u>                                                             | CM scheme . . . . . <u>*TURNOVER</u>    |  |  |
| Description . . . . .                                       | <u>SQLSTM- TRUNSQLSTM</u>                                                 | Attribute . . . . . <u>SQL</u>          |  |  |
| Object type . . . . .                                       | <u>*SQLSTM</u>                                                            | Source file name . . . <u>QSQLSRC</u>   |  |  |
| Sequence . . . . .                                          | <u>800</u>                                                                | Data object . . . . . <u>Y *DEFAULT</u> |  |  |
| Create command . . .                                        | <u>SOFTTURN/TRUNSQLSTM SRCFILE("&amp;SL"/"&amp;SF") SRCMBR("&amp;SM")</u> |                                         |  |  |
| <u>COMMIT (*NONE) DFTRDBCOL("&amp;LI") RUNAS("&amp;U9")</u> |                                                                           |                                         |  |  |
|                                                             |                                                                           |                                         |  |  |

This command **MUST** be qualified when used in the type code definition because it resides in the language library, which typically is not in the library list during a TURNOVER® Lifecycle Manager form run.

This command builds and executes the `RUNSQLSTM` command from within a program that does not use adopted authority. In this way, you can override the adopted authority present during a TURNOVER® Lifecycle Manager form. There are these additional parameters:

### Run as user profile (RUNAS)

Indicates the user profile under whose authority settings you want the `RUNSQLSTM` command to run.

Use a type code substitution variable for this parameter at the different levels of your application. At lower levels of the application, use a value of `*CURRENT`; at higher levels, specify a value such as `QSECOFR` or the application owner.

Add an entry for the `RUNAS` keyword into the `SOFTTURN/TKEYPRTF` file. This file is where you record command keyword entries that should never be overridden. The entry in the `TKEYPRTF` file should look like this:

```
KKTYPE: *ALL
KKKEY:  RUNAS
```

This entry ensures that `RUNAS (*CURRENT)` cannot be overridden at the lower levels of your application.

### Extra Default Parm (EXTRADFT)

When defining the `TRUNSQLSTM` command on a global type code, use this field to include any additional `RUNSQLSTM` command parameter values that should be treated as global defaults.

### Extra User-Specified Parm (EXTRAUSER)

When a programmer prompts the `TRUNSQLSTM` command in TURNOVER® Lifecycle Manager at compile time, they should use this field to supply any additional non-global, `RUNSQLSTM` command parameter values that should be used for the compile.

---

## Distributing SQL changes

When distributing SQL, you must “compile” on all systems to execute the SQL script.

These instructions assume that you are just distributing objects to the remote and using `MO` (Move Object) as your promotion method. If you are already distributing source code and compiling on the remote, you might already be setup correctly.

- 1 Ensure that every SQL type code in the distributing level of the sending application is set to distribute source. Most applications that distribute have the **Distribute Source** parameter set to "N," so you must edit each SQL type code in the distributing application level and make sure this setting is "Y."
- 2 On the remote (production) computer, have TURNOVER® Lifecycle Manager store the source that it receives into library QTEMP. To do this, change the **Source Target Library** value to QTEMP in the corresponding level of the remote application definition that is used to receive and run the remote form. This deletes the source code from the Production machine after the form completes and it is no longer needed for the creation of the object.
- 3 If you prefer to leave a copy of the scripts on the remote system, specify a source file and library instead of QTEMP.
- 4 Override every SQL type code in the remote application definition to use method `CSCO`. This forces the recreation of the objects from the source. The IBM SQL licensed programs must be installed on the production machine as well. This recommendation ensures that TURNOVER® Lifecycle Manager accurately creates the objects on the target computer, maintaining all constraints and triggers exactly as they were defined and named on the development machine.
- 5 On the remote computer, verify that your application library list is correct for compiling.

---

## Object creation sequence

Your work in development (for example, from the worklist) is largely up to you. You decide what should compile first (the parent or base table), and what second (the child or related table, views and indexes). If you were creating the objects in the examples below for the first time in development, you need to compile the objects in the following order:

- 1 Compile CUSTOMER and PART first because they are the parent files.
- 2 Compile SALE next because of the foreign key constraint on CUSTOMER.
- 3 Compile CUSTERR next because it is the logical view over CUSTOMER.
- 4 Compile SALEITEM next because it has foreign keys over both CUSTOMER and SALE.
- 5 Compile CUSSALPRT last because it is a logical view over CUSTOMER, SALE, and PART.

When you promote these items to QA, TURNOVER® Lifecycle Manager looks at the compile dates and times of the objects and orders them on the form according to that (within type code, of course). This makes the form "smart" enough to create the items in the proper order.

## Example: DDL source for a simple database

This is an example of some DDL source members. Each source member is separated by =====. The name of the source member must be the same as the name of the object it creates.

```
=====
Create Table Customer
( CustID      Dec(    7, 0 ) Not Null,
  Name        Char(   30 ) Not Null,
  ShpLine1    Char(  100 ) Not Null,
  ShpLine2    Char(  100 ) Not Null,
  ShpCity     Char(   30 ) Not Null,
  ShpState    Char(    2 ) Not Null,
  ShpPsCd1    Char(   10 ) Not Null,
  ShpPsCd2    Char(   10 ) Not Null,
  ShpCntry    Char(   30 ) Not Null,
  PhnVoice    Char(   15 ) Not Null,
  PhnFax      Char(   15 ) Not Null,
  PhnExt      Char(    4 ) Not Null,
  Territory   Char(    1 ) Not Null,
  Status      Char(    1 ) Not Null
              With Default ' ',
  CrdLimit    Dec(    7, 0 ) With Default Null,
  EntDate     Date          Not Null,
Primary Key( CustID ),
Constraint StatusChk
  Check(Status IN ( ' ', '1', '2', '3' )),
Constraint TerritoryChk
  Check(Territory IN ( '1', '2', '3' )),
Constraint CreditChk
  Check( CrdLimit >= 0 ) )
=====

Create Table Part
( PartID      Dec(    7, 0 ) Not Null,
  PartDesc    Char(   50 ) Not Null,
Primary Key( PartID ) )
=====

Create Table Sale
( OrderID     Dec(    7, 0 ) Not Null,
  SaleDate    Date          Not Null,
  SaleTot     Dec(    7, 2 ) Not Null,
  CustID      Dec(    7, 0 ) Not Null,
  TotPrice    Dec(    7, 2 ) Not Null,
Primary Key( OrderID ),
Constraint SalesCustFK Foreign Key( CustID )
  References Customer( CustID )
  On Delete Cascade )
=====

Create Table SaleItem
( OrderID     Dec(    7, 0 ) Not Null,
  PartID      Dec(    7, 0 ) Not Null,
  Quantity    Dec(    7, 0 ) Not Null,
  Price       Dec(    7, 2 ) Not Null,
  Discount    Dec(    7, 2 ) Not Null,
Primary Key( OrderID, PartID ),
Constraint DiscountChk
  Check( Discount <= Price ),
Constraint SlItmOrdFK Foreign Key( OrderID )
  References Sale( OrderID )
  On Delete Cascade
  On Update Restrict,
Constraint SlItmPrtFK Foreign Key( PartID )
  References Part( PartID )
  On Delete Restrict
  On Update Restrict );

Grant Select,
      Update( Quantity,
              Price,
              Discount)
```



```
        On SaleItem
        To GrpAcctng;

Grant Select,
        Update( Quantity,
                Discount)
        On SaleItem
        To GrpSales;
=====

Create Index CustTerr
On Customer
( Territory,
  Name )
=====

Create View CusSalPrt
( CustID,
  Name,
  SaleDate,
  PartDesc )
As Select Customer.CustID,
        Name,
        SaleDate,
        PartDesc
From   Customer,
        Sale,
        SaleItem,
        Part
Where  Customer.CustID = Sale.CustID
      And Sale.OrderID = SaleItem.OrderID
      And SaleItem.PartID = Part.PartID
```

## 24 Managing disk space

If you are short of DASD space, here are some things that you can do to free space after installing TURNOVER® Lifecycle Manager.

---

### Delete the TURNOVER® Lifecycle Manager installation library

After you have installed TURNOVER® Lifecycle Manager on all of your systems, you can delete the entire install library. Its size varies depending on the release of TURNOVER® you have installed.

| Release                          | Library   |
|----------------------------------|-----------|
| TURNOVER® Lifecycle Manager v2.n | TLM2XINST |
| TURNOVER® for iSeries v101       | T101INST  |
| TURNOVER® for iSeries v100       | T100INST  |
| TurnOver v5.4                    | T100INST  |
| TurnOver v5.3                    | T53INST   |

You must always restore this library each time you receive a new installation package. The only thing in this library that you might want to keep is the listing of all of the PTFs. If you want to retain these online, *delete* all the save files in this library.

**Note** Do *not* clear the save files; *delete* them. The installation programs check that the save files exist before they proceed. Therefore, if these files do not exist, you cannot accidentally install the wrong objects.

---

### Delete temporary TURNOVER® Lifecycle Manager libraries

TURNOVER® Lifecycle Manager creates and uses temporary libraries during form and distribution processing. These include:

- TnnnnnnnD/Dn libraries

- `TnnnnnnnXX` libraries
- `TnnnnnnnS` libraries (where `nnnnnnn` is the form number).

Libraries `TnnnnnnnD` (for ASP1) and `TnnnnnnnDn` (for ASPs other than 1), as well as `TnnnnnnnXX` libraries, are created and used during a form run to manage objects that will be promoted and exploded, respectively. A `TnnnnnnnS` library is created during distribution to manage objects that will be sent to remote systems; it has different significance on remote systems than on development systems.

If you have defined your application to “explode” promoted objects to additional locations, a `TnnnnnnnXX` library is created where `XX` represents a number, starting at 01. (The number of explode libraries you have determines how many of the `TnnnnnnnXX` libraries are created.) If you are not archiving objects *and* the form finishes with a status of `RAN-OK`, these libraries are deleted automatically; otherwise, they stay on your system.

---

## Deleting temporary archive and explode libraries

To remove the `TnnnnnnnD` and `TnnnnnnnXX` libraries, you should first confirm for each library that the associated form is not currently running; then you can delete them. Alternatively, you can run the `SAVARSBM` command, which saves these libraries to tape and then deletes them. This job tells TURNOVER® Lifecycle Manager Object History what tape volume the libraries are on, so that if you ever need to restore one of them, you can do so.

---

## Deleting temporary distribution libraries

To delete `TnnnnnnnS` libraries:

- On a *local development computer* that does not receive distributions:  
  
Check for libraries associated with forms that are distributing at that moment. You can safely delete any others.
- On a *remote computer*  
  
Check to make sure these libraries contain objects for forms that have been received, but not run. Any others can safely be deleted.

---

## Manage source archive files

Use option 2 on the TURNOVER® Lifecycle Manager Utility menu to review the source archive files that are in use on your system. Save any archive files that need not remain online off to tape, and then delete them.

---

## Manage IFS archive files

To manage IFS archive files, use main menu option 8, then option 14.

---

## Check for old upgrade backup libraries

Look for old versions of TURNOVER® Lifecycle Manager that were replaced by an upgrade, and delete the libraries. These libraries are named TOLNG0001, TOPRD0001, and TODTA0001.

---

## Still tight on disk space?

If you are still tight on DASD after performing the steps outlined above, consider whether an IBM i disk management solution, such as SoftLanding's SUPERMON® Disk, would be of benefit to you.

# A Job descriptions

---

## Job descriptions

A job description enables you to create a named set of *job properties* that you can save and reuse. Job properties tell TURNOVER® Lifecycle Manager how to handle a job, including when to start it, where to get the job from, and how the job will run.

TURNOVER® Lifecycle Manager uses its own job descriptions for most of the work it does. This topic lists the different job descriptions TURNOVER® Lifecycle Manager uses.

| Job description | Used by                                                                                                                                                                                                                                                                                                      |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| HELPDESK        | <p>TURNOVER® Client jobs run under the HELPDESK job description by default. Used to determine the product library at the time of configuring autostart jobs.</p> <p>See <a href="#">‘HELPDESK job description’ on page 394.</a></p>                                                                          |
| TOMSGRCV        | <p>Used by the TOMSGRCV autostart job to log form activity and confirmations.</p> <p>See <a href="#">‘TOMSGRCV job description’ on page 395.</a></p>                                                                                                                                                         |
| TURCRARE        | <p>Used on production systems by jobs that are automatically receiving and submitting TURNOVER® Lifecycle Manager distributions via SNADS from a development system. (This profile does not apply if you are using FTP distribution.)</p> <p>See <a href="#">‘TURCRARE job description’ on page 397.</a></p> |
| TURNESCQ        | <p>Used by the TURNESCQ job for task escalation and related messaging.</p> <p>See <a href="#">‘TURNESCQ job description’ on page 398.</a></p>                                                                                                                                                                |
| TURNOVER        | <p>Default job description used for general TURNOVER® Lifecycle Manager operations, such as reporting and distribution.</p> <p>See <a href="#">‘TURNOVER job description’ on page 399.</a></p>                                                                                                               |

---

| Job description | Used by                                                                                                                                           |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| WDAUTO          | Used by the WISEDESK autostart job that monitors the Wisedesk dynamic update data queue.<br><br>See <i>'WDAUTO job description' on page 401.</i>  |
| WDDYNUPD        | Used by the WDDYNUPD command, which starts, stops, or delays Wisedesk dynamic updating.<br><br>See <i>'WDDYNUPD job description' on page 402.</i> |
| TMAILD          | Used by the TURNOVER® Lifecycle Manager mail daemon to process email escalation messaging.                                                        |

## HELPDESK job description

TURNOVER® Client jobs run under the HELPDESK job description by default.

| Attribute | Value         |
|-----------|---------------|
| JOBID     | QGPL/HELPDESK |
| USER      | *RQD          |
| SYNTAX    | *NOCHK        |
| HOLD      | *NO           |
| ENDSEV    | 30            |
| DATE      | *SYSVAL       |
| SWS       | 00000000      |
| INQMSGRPY | *RQD          |
| JOBPTY    | 5             |
| JOBQ      | QGPL/QINTER   |
| OUTPTY    | 5             |
| PRTDEV    | *USRPRF       |
| OUTQ      | *USRPRF       |
| LOG       |               |
| Level     | 4             |

| Attribute | Value                           |
|-----------|---------------------------------|
| Severity  | 0                               |
| Text      | *NOLIST                         |
| LOGCLPGM  | *NO                             |
| ACGCDE    | *USRPRF                         |
| PRTTXT    | *SYSVAL                         |
| RTGDTA    | QCMDI                           |
| RQSDTA    |                                 |
| DDMCNV    | *KEEP                           |
| DEVRCYACN | *SYSVAL                         |
| TSEPOOL   | *SYSVAL                         |
| JOBMSGQMX | *SYSVAL                         |
| MOBMSGQFL | *WRAP                           |
| ALWMLTTHD | *NO                             |
| INLASGRP  | *NONE                           |
| SPLFACN   | *SYSVAL                         |
| TEXT      | TURNOVER Client Job Description |

## TOMSGRCV job description

This job description is used by the TOMSGRCV autostart job to log form activity and confirmations.

| Attribute | Value         |
|-----------|---------------|
| JOBD      | QGPL/TOMSGRCV |
| USER      | TOMSGRCV      |
| SYNTAX    | *NOCHK        |
| HOLD      | *NO           |
| ENDSEV    | 30            |

| Attribute | Value         |
|-----------|---------------|
| DATE      | *SYSVAL       |
| SWS       | 00000000      |
| INQMSGRPY | *RQD          |
| JOBPTY    | 5             |
| JOBQ      | QGPL/QINTER   |
| OUTPTY    | 5             |
| PRTDEV    | *USRPRF       |
| OUTQ      | *USRPRF       |
| LOG       |               |
| Level     | 4             |
| Severity  | 0             |
| Text      | *NOLIST       |
| LOGCLPGM  | *NO           |
| ACGCDE    | *USRPRF       |
| PRTTXT    | *SYSVAL       |
| RTGDTA    | QCMDI         |
| RQSDTA    | CALL TURCRCVA |
| DDMCNV    | *KEEP         |
| DEVRCYACN | *SYSVAL       |
| TSEPOOL   | *SYSVAL       |
| JOBMSGQMX | *SYSVAL       |
| MOBMSGQFL | *SYSVAL       |
| ALWMLTTHD | *NO           |
| INLASGRP  | *NONE         |
| SPLFACN   | *SYSVAL       |



| Attribute | Value |
|-----------|-------|
| TEXT      |       |

## TURCRARE job description

The TURCRARE job description is used on production systems by jobs that are automatically receiving and submitting TURNOVER® Lifecycle Manager distributions from a development system, via SNADS.

| Attribute | Value         |
|-----------|---------------|
| JOBID     | QGPL/TURCRARE |
| USER      | TURNOVER      |
| SYNTAX    | *NOCHK        |
| HOLD      | *NO           |
| ENDSEV    | 30            |
| DATE      | *SYSVAL       |
| SWS       | 00000000      |
| INQMSGRPY | *RQD          |
| JOBPTY    | 5             |
| JOBQ      | QGPL/QINTER   |
| OUTPTY    | 5             |
| PRTDEV    | *USRPRF       |
| OUTQ      | *USRPRF       |
| LOG       |               |
| Level     | 4             |
| Severity  | 0             |
| Text      | *NOLIST       |
| LOGCLPGM  | *NO           |
| ACGCDE    | *USRPRF       |

| Attribute | Value                                              |
|-----------|----------------------------------------------------|
| PRTTXT    | *SYSVAL                                            |
| RTGDTA    | QCMDI                                              |
| RQSDTA    | CALL PGM(TURCRARE)                                 |
| DDMCNV    | *KEEP                                              |
| DEVRCYACN | *SYSVAL                                            |
| TSEPOOL   | *SYSVAL                                            |
| JOBMSGQMX | *SYSVAL                                            |
| MOBMSGQFL | *SYSVAL                                            |
| ALWMLTTHD | *NO                                                |
| INLASPGRP | *NONE                                              |
| SPLFACN   | *SYSVAL                                            |
| TEXT      | TURNOVER auto-received JOBID for production system |

## TURNESCQ job description

This job description is used by the TURNESCQ job for task escalation and related messaging.

| Attribute | Value         |
|-----------|---------------|
| JOBID     | QGPL/TURNESCQ |
| USER      | TURNESCQ      |
| SYNTAX    | *NOCHK        |
| HOLD      | *NO           |
| ENDSEV    | 30            |
| DATE      | *SYSVAL       |
| SWS       | 00000000      |
| INQMSGRPY | *RQD          |
| JOBPTY    | 5             |
| JOBQ      | QGPL/QINTER   |

| Attribute | Value                                                 |
|-----------|-------------------------------------------------------|
| OUTPTY    | 5                                                     |
| PRTDEV    | *USRPRF                                               |
| OUTQ      | *USRPRF                                               |
| LOG       |                                                       |
| Level     | 4                                                     |
| Severity  | 0                                                     |
| Text      | *MSG                                                  |
| LOGCLPGM  | *YES                                                  |
| ACGCDE    | *USRPRF                                               |
| PRTTXT    | *SYSVAL                                               |
| RTGDTA    | QCMDB                                                 |
| RQSDTA    | TSTRESCQ                                              |
| DDMCNV    | *KEEP                                                 |
| DEVRCYACN | *SYSVAL                                               |
| TSEPOOL   | *SYSVAL                                               |
| JOBMSGQMX | *SYSVAL                                               |
| MOBMSGQFL | *SYSVAL                                               |
| ALWMLTTHD | *NO                                                   |
| INLASPGRP | *NONE                                                 |
| SPLFACN   | *SYSVAL                                               |
| TEXT      | JobD for TURNOVER® Lifecycle Manager Escalation Queue |

## TURNOVER job description

This job description is used for general TURNOVER® Lifecycle Manager operations, such as reporting and distribution.

| Attribute | Value             |
|-----------|-------------------|
| JOBID     | SOFTTURN/TURNOVER |
| USER      | *RQD              |
| SYNTAX    | *NOCHK            |
| HOLD      | *NO               |
| ENDSEV    | 30                |
| DATE      | *SYSVAL           |
| SWS       | 00000000          |
| INQMSGRPY | *RQD              |
| JOBPTY    | 5                 |
| JOBQ      | QGPL/QBATCH       |
| OUTPTY    | 5                 |
| PRTDEV    | *USRPRF           |
| OUTQ      | TURNOVER          |
| LOG       |                   |
| Level     | 4                 |
| Severity  | 0                 |
| Text      | *NOLIST           |
| LOGCLPGM  | *NO               |
| LOGOUTPUT | *SYSVAL           |
| ACGCDE    | *USRPRF           |
| PRTTXT    | *SYSVAL           |
| RTGDTA    | QCMDI             |
| RQSDTA    | *NONE             |
| DDMCNV    | *KEEP             |
| DEVRCYACN | *SYSVAL           |

| Attribute | Value                    |
|-----------|--------------------------|
| TSEPOOL   | *SYSVAL                  |
| JOBMSGQMX | *SYSVAL                  |
| MOBMSGQFL | *SYSVAL                  |
| ALWMLTTHD | *NO                      |
| INLASPRG  | *NONE                    |
| SPLFACN   | *SYSVAL                  |
| TEXT      | TURNOVER Job Description |

## WDAUTO job description

This job description is used by the WISEDESK autostart job that monitors the Wisedesk dynamic update data queue.

| Attribute | Value           |
|-----------|-----------------|
| JOBID     | SOFTTURN/WDAUTO |
| USER      | TURNESCQ        |
| SYNTAX    | *NOCHK          |
| HOLD      | *NO             |
| ENDSEV    | 30              |
| DATE      | *SYSVAL         |
| SWS       | 00000000        |
| INQMSGRPY | *RQD            |
| JOBPTY    | 5               |
| JOBQ      | QGPL/QINTER     |
| OUTPTY    | 5               |
| PRTDEV    | *USRPRF         |
| OUTQ      | *USRPRF         |
| LOG       |                 |

| Attribute | Value                                   |
|-----------|-----------------------------------------|
| Level     | 4                                       |
| Severity  | 0                                       |
| Text      | *NOLIST                                 |
| LOGCLPGM  | *NO                                     |
| LOGOUTPUT | *SYSVAL                                 |
| ACGCDE    | *USRPRF                                 |
| PRTTXT    | *SYSVAL                                 |
| RTGDTA    | QCMDI                                   |
| RQSDTA    | WDAUTO                                  |
| DDMCNV    | *KEEP                                   |
| DEVRCYACN | *SYSVAL                                 |
| TSEPOOL   | *SYSVAL                                 |
| JOBMSGQMX | *SYSVAL                                 |
| MOBMSGQFL | *SYSVAL                                 |
| ALWMLTTHD | *NO                                     |
| INLASGRP  | *NONE                                   |
| SPLFACN   | *SYSVAL                                 |
| TEXT      | Wisedesk Dynamic Updating Autostart Job |

## WDDYNUPD job description

This job description is used by the WDDYNUPD command, which starts, stops, or delays Wisedesk dynamic updating.

| Attribute | Value             |
|-----------|-------------------|
| JOBID     | SOFTTURN/WDDYNUPD |
| USER      | *RQD              |
| SYNTAX    | *NOCHK            |

| Attribute | Value       |
|-----------|-------------|
| HOLD      | *NO         |
| ENDSEV    | 30          |
| DATE      | *SYSVAL     |
| SWS       | 00000000    |
| INQMSGRPY | *RQD        |
| JOBPTY    | 5           |
| JOBQ      | QGPL/QINTER |
| OUTPTY    | 5           |
| PRTDEV    | *USRPRF     |
| OUTQ      | *USRPRF     |
| LOG       |             |
| Level     | 4           |
| Severity  | 0           |
| Text      | *NOLIST     |
| LOGCLPGM  | *NO         |
| ACGCDE    | *USRPRF     |
| PRTTXT    | *SYSVAL     |
| RTGDTA    | QCMDI       |
| RQSDTA    | *NONE       |
| DDMCNV    | *KEEP       |
| DEVRCYACN | *SYSVAL     |
| TSEPOOL   | *SYSVAL     |
| JOBMSGQMX | *SYSVAL     |
| MOBMSGQFL | *SYSVAL     |
| ALWMLTTHD | *NO         |

| Attribute | Value                                   |
|-----------|-----------------------------------------|
| INLASPGRP | *NONE                                   |
| SPLFACN   | *SYSVAL                                 |
| TEXT      | Wisedesk Dynamic Updating Autostart Job |



## B Configuration checklists

### Getting started

| Task                                                                                                                        | Done                     |
|-----------------------------------------------------------------------------------------------------------------------------|--------------------------|
| Change the company name field.                                                                                              | <input type="checkbox"/> |
| <i>In the Client:</i> Expand <b>Administration &gt; Utilities</b> , and then click <b>Global Defaults – Miscellaneous</b> . |                          |
| <i>On the IBM i computer:</i> Select Main Menu option 8, and then select option 5.                                          |                          |
| Who will be assigned as the TURNOVER® Lifecycle Manager Administrator?                                                      | <input type="checkbox"/> |
| What applications need to be defined first?                                                                                 | <input type="checkbox"/> |
| Which TURNOVER® Lifecycle Manager application model meets you software application's requirements best?                     | <input type="checkbox"/> |
| Repeat this evaluation for each software application you will manage.                                                       |                          |
| What TURNOVER® Lifecycle Manager projects will you need?                                                                    | <input type="checkbox"/> |
| See "Working with projects and tasks" in the <i>TURNOVER® Lifecycle Manager User Guide</i> .                                |                          |
| Which users will be changing the application?                                                                               | <input type="checkbox"/> |
| Which users will have access to the project system?                                                                         | <input type="checkbox"/> |
| To which systems will you be distributing changes for this software application?                                            | <input type="checkbox"/> |

### General setup

| Task                                                                           | Done                     |
|--------------------------------------------------------------------------------|--------------------------|
| Enroll and authorize TURNOVER® Lifecycle Manager users (main menu option 3)    | <input type="checkbox"/> |
| See "Authorizing users" in the <i>TURNOVER® Lifecycle Manager User Guide</i> . |                          |

| Task                                                                                                                                                                                                                                                                                                          | Done                     |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| Enroll programmers (main menu option 2).<br><br>See "Setting up programmer profiles" in the <i>TURNOVER® Lifecycle Manager User Guide</i> .                                                                                                                                                                   | <input type="checkbox"/> |
| Define the computers in your network (main menu option 11, and then option 1).                                                                                                                                                                                                                                | <input type="checkbox"/> |
| Install TURNOVER® Lifecycle Manager on your remote systems.<br><br>See "Preparing to install, upgrade, or apply changes" in the <i>TURNOVER® Lifecycle Manager Installation</i> manual.                                                                                                                       | <input type="checkbox"/> |
| Copy the application models that best suits your needs.<br><br><i>In the Client:</i> Right-click an application definition, and then click <b>Copy Application</b> .<br><br><i>On the IBM i computer:</i> Main menu option 1, option 3.                                                                       | <input type="checkbox"/> |
| Change your custom model to match your requirements.<br><br><i>In the Client:</i> Right-click an application definition, and then click <b>Checkout Application</b> .<br><br><i>On the IBM i computer:</i> Main Menu option 1, option 2.                                                                      | <input type="checkbox"/> |
| Create your application definition by copying your model.<br><br><i>In the Client:</i> Right-click your custom model, and then click <b>Copy application</b> .<br><br><i>On the IBM i computer:</i> Main Menu option 1, then option 3.<br><br>See <a href="#">'Copying an application model' on page 37</a> . | <input type="checkbox"/> |
| Define the TURNOVER® Lifecycle Manager projects.<br><br>Use either TURNOVER® Client Projects, or main menu option 9, then F6.<br><br>See "Working with projects and tasks" in the <i>TURNOVER® Lifecycle Manager User Guide</i> .                                                                             | <input type="checkbox"/> |
| Create test objects with which to test your setup.                                                                                                                                                                                                                                                            | <input type="checkbox"/> |
| Test each object type using your application definition.                                                                                                                                                                                                                                                      | <input type="checkbox"/> |
| Create a task, build a worklist, add test objects to the worklist, check out objects, build and run TURNOVER® Lifecycle Manager forms, distribute, and install changes (if appropriate).                                                                                                                      | <input type="checkbox"/> |

## Distribution setup

### On the development computer

| Task                                                                                                                                                                                                                                                   | Done                     |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| Define all the TURNOVER® Lifecycle Manager computers in your network, including the development computer.                                                                                                                                              | <input type="checkbox"/> |
| Define the Send distribution defaults for each production computer.                                                                                                                                                                                    | <input type="checkbox"/> |
| <i>In the Client:</i> Change a distribution system, and then go to the Send Defaults page.                                                                                                                                                             |                          |
| <i>On the IBM i computer:</i> Main menu 11, option 1, and then select a system with option 9.                                                                                                                                                          |                          |
| (If any type of "source" is to be distributed for any application, you must enter a Y in the distribute source field in that computer's Send defaults. You will be able to restrict source distribution by application or by object type, if desired.) | <input type="checkbox"/> |
| Verify the operating system levels of all computers involved and set the TGTRLS global default value on the development computer accordingly. (You can override this value for a specific application if necessary.)                                   | <input type="checkbox"/> |
| <i>In the Client:</i> Go to <b>Administration &gt; Utilities</b> , and then double-click <b>Global Defaults</b> . The Target Release parameter is on the Form-Related page.                                                                            |                          |
| <i>On the IBM i computer:</i> Main menu option 8, then option 5.                                                                                                                                                                                       |                          |
| Verify that the message receiver autostart job (TOMSGRCV) is running in subsystem.                                                                                                                                                                     | <input type="checkbox"/> |
| Autostart job TOMSGRCV must be running on the development system. This job is set up for you during installation. It waits for a confirmation message to be sent from the production computers, and then updates the form status there.                | <input type="checkbox"/> |
| Create a remote application definition for each distributing application level, and distribute it to each production computer.                                                                                                                         | <input type="checkbox"/> |
| Do this now only if you have both created your application definitions <i>and</i> set up your distribution defaults. If you don't do this now, you can come back and do it later.                                                                      |                          |

### On each production computer

| Task                                                                                                                              | Done                     |
|-----------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| Create system definitions for the development computer and the production computer.                                               | <input type="checkbox"/> |
| <i>In the Client:</i> Expand <b>Administration</b> , right-click <b>Distribution Systems</b> , and then click <b>New System</b> . |                          |
| <i>On the IBM i computer:</i> Main Menu option 11, then option 1.                                                                 |                          |

| Task                                                                                                                                                                                                                                                                                                                                              | Done                     |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| Define the Receive distribution defaults for the local production system.                                                                                                                                                                                                                                                                         | <input type="checkbox"/> |
| The target user must be the same as that defined on the development system.                                                                                                                                                                                                                                                                       | <input type="checkbox"/> |
| Verify that the TURCRARE autostart job is running in subsystem TSERVER.                                                                                                                                                                                                                                                                           | <input type="checkbox"/> |
| This job is set up and started in subsystem TSERVER during installation. The TURCRARE autostart job must be running on each production computer. This job waits for a distribution to be sent to the target user; receives the distribution file; and then checks the RECEIVE distribution defaults to determine if the form should be submitted. |                          |
| Review the global type codes on each production system.                                                                                                                                                                                                                                                                                           | <input type="checkbox"/> |
| If you have created any additional global type codes on the development system, also create them on the production systems                                                                                                                                                                                                                        | <input type="checkbox"/> |
| Review the application definition on each production system                                                                                                                                                                                                                                                                                       | <input type="checkbox"/> |
| In your remote application definition, you will probably want to change <b>Method</b> to MO if you are distributing objects only, <b>Require source checkout</b> to N, and <b>Require project entry</b> to N.                                                                                                                                     | <input type="checkbox"/> |

# C Commands

This section describes TURNOVER® Lifecycle Manager commands:

- TURNOVER® Lifecycle Manager main menu and submenu commands
- commands that are not associated with any TURNOVER® Lifecycle Manager user interface.

You should run some of these commands from a command line and some as PDM user-defined options.

## Main menu options

You can call TURNOVER® Lifecycle Manager main menu options directly, or you can set them up as options in SoftMenu (or in menus you create).

| Option | Description                       | Command           |
|--------|-----------------------------------|-------------------|
| 1      | Work with application definitions | TURNOVER OPT (1)  |
| 2      | Maintain programmer information   | TURNOVER OPT (2)  |
| 3      | View Users authority              | TURNOVER OPT (3)  |
| 4      | Work with object history          | TURNOVER OPT (4)  |
| 5      | Check-out/check-in menu           | TURNOVER OPT (5)  |
| 6      | Work with forms                   | TURNOVER OPT (6)  |
| 7      | Submit a form to run              | TURNOVER OPT (7)  |
| 8      | Utility menu                      | TURNOVER OPT (8)  |
| 9      | Work with Projects and Tasks      | TURNOVER OPT (9)  |
| 10     | Reporting menu                    | TURNOVER OPT (10) |
| 11     | Distribution menu                 | TURNOVER OPT (11) |

| Option | Description                                                                                           | Command |
|--------|-------------------------------------------------------------------------------------------------------|---------|
| 12     | User-defined option (depends on contents of data area SOFTTURND/CASECMD). Set to STRPWM when shipped. |         |

## Submenus

### Checkout/check-in menu

| Option | Description                       | Command    |
|--------|-----------------------------------|------------|
| 1      | View checked out objects/members  | DSPCHKOUT  |
| 2      | Check-in by object                | CHKINOBJ   |
| 3      | Check-in by member                | CHKINMBR   |
| 4      | Checkout                          | CHECKOUT   |
| 10     | View checked out IFS objects      | No command |
| 11     | Check-in IFS object               | TCHKINIFS  |
| 12     | Checkout IFS object               | TCHKOUTIFS |
| 20     | Work with object references       | WRKOBJREF  |
| 21     | Work with user-defined references | WRKUSRREF  |
| 22     | Programmer Worklist Manager       | STRPWM     |
| 30     | Work with *DATA definitions       | WRKTODATA  |
| 31     | Checkout S/36 Menu                | CHKOUTMEN  |
| 32     | Checkout S/36 DFU                 | CHKOUTDFU  |

### Utility menu

| Option | Description                     | Command  |
|--------|---------------------------------|----------|
| 1      | Submit file purge               | PURFOR   |
| 2      | Work with source archive files  | SAVARCS  |
| 4      | Work with type code definitions | WRKTOTYP |
| 5      | Set global defaults             | SETGLOBE |

| Option | Description                                 | Command    |
|--------|---------------------------------------------|------------|
| 6      | Display current release level               | DSPRELTO   |
| 7      | Work with exits                             | WRKTOEXIT  |
| 8      | Submit reorganize of files                  | RGZTOFILES |
| 9      | Submit save of archived objects             | SAVARCO    |
| 10     | Lookup archived objects                     | No command |
| 11     | Set source compare defaults                 | No command |
| 12     | Global Programmer Worklist session defaults | No command |
| 13     | Work with type code conversion table        | TCVTOBJTYP |
| 14     | Submit IFS archive save                     | TSAVIFSARC |

## Reporting menu

| Option | Description                    | Command    |
|--------|--------------------------------|------------|
| 1      | Print application definition   | LISTDFTS   |
| 2      | Check application definition   | CHKAPPDFN  |
| 3      | Audit report                   | AUDITRPT   |
| 4      | Audit adjustments and report   | AUDITADJ   |
| 10     | Projects and tasks             | PRJTSKRPT  |
| 11     | Resource time report           | PRJRSRTIM  |
| 12     | Project timesheet report       | TIMERPT    |
| 13     | Projects and tasks by priority | PRJPTYRPT  |
| 14     | Project cost report            | PRJCSTRPT  |
| 15     | Task aging report              | TSKAGERPT  |
| 16     | Project definition report      | PRJDEFRPT  |
| 20     | List Forms                     | LISTFORMS  |
| 21     | Object history                 | RPTOBJHST  |
| 22     | Programmer history             | RPTPGMRHST |

| Option | Description                         | Command    |
|--------|-------------------------------------|------------|
| 23     | Source checkout history             | RPTSRCUSE  |
| 24     | Reference activity report           | REFACTRPT  |
| 25     | Related application conflict report | TRPTCON    |
| 30     | Form distribution report            | RPTFORMDST |
| 31     | Object distribution report          | RPTOBJDST  |
| 40     | List authorized users               | LISTUSERS  |
| 41     | Programmer profile information      | LISTPGMRS  |

### Distribution menu

| Option | Description                              | Command    |
|--------|------------------------------------------|------------|
| 1      | Maintain system definitions              | WRKDSTSYS  |
| 2      | Distribute form                          | DSTFOR     |
| 3      | Receive Distribution                     | RCVFOR     |
| 4      | Work with form log                       | WRKFORMLOG |
| 5      | Work with remote application definitions | WRKRMTAPP  |
| 6      | Override network distribution method     | TCFGNETDST |
| 7      | Remote Configuration Wizard              | TRMTCFGWIZ |

### Additional menu commands

| Description                                 | Command    |
|---------------------------------------------|------------|
| Access Helpdesk Requester panel             | HELPDESK   |
| Access Helpdesk request entry panel         | FASTTASK   |
| Access work task coordinator's review panel | WORKTASK   |
| Access Wisedesk help trees                  | WRKHLPTREE |



---

## TURNOVER® Lifecycle Manager commands

### **ADDCATSTR: Add category search string**

Add search strings to a category or subcategory's task selection criteria.

For more information about help trees, see "Wisedesk" in the *TURNOVER® Lifecycle Manager User Guide*.

### **ADDHLPCAT: Add help tree category**

Add a category or subcategory to a help tree.

For more information about help trees, see "Wisedesk" in the *TURNOVER® Lifecycle Manager User Guide*.

### **ADDHLPTREE: Add help tree**

This command is executed when you use option 1 on the Work with Help Trees panel, to create a new help tree on your system.

For more information about help trees, see "Wisedesk" in the *TURNOVER® Lifecycle Manager User Guide*.

### **ADDPWLITM: Add programmer worklist item**

Add an item (object or source member name) to a programmer's worklist during the preliminary stages of a project task.

When you specify \*ACTIVE value for the **List** parameter, the item is added to the active worklist; if there is no active worklist, the item is added to the worklist that was last active.

### **ADDTIMSHT: Add timesheet entry**

Add a timesheet entry to a resource's online timesheet.

### **AUDITADJ: Application audit adjustments and report**

Adjust TURNOVER® Lifecycle Manager's audit control file.

See "Reports" in the *TURNOVER® Lifecycle Manager User Guide*.

### **AUDITRPT: Application audit report**

Print any object or source member changed outside of TURNOVER® Lifecycle Manager's scope of control for the specified application.

### **BLDHLPCAT: Build help tree category**

Builds a help tree category.

This command is the same as selecting a help tree, category, or subcategory with option 14=Build. (The contents of the tree and path fields vary depending on which one you select.)

For more information about help trees, see “Wisedesk” in the *TURNOVER® Lifecycle Manager User Guide*.

### **CHECKOUT: Check out source**

For both batch and interactive checkout processing from the Programmer Worklist (using option 21).

You can also use this command from the TURNOVER® Lifecycle Manager command line to check out source by object or by source member.

### **CHGCATSTR: Change category search string**

Change an existing search string in a category’s task selection criteria. This is the same as using option 2=Change on the Work with Category Search Strings panel.

### **CHGHLPCAT: Change help tree category**

Change the task selection criteria for an existing help tree category.

This command is the same as selecting either a help tree category or subcategory with option 2=Change. (The tree and path fields vary depending on which one you select.)

For more information about help trees, see “Wisedesk” in the *TURNOVER® Lifecycle Manager User Guide*.

### **CHGHLPTREE: Change help tree**

Change the description of a help tree. This command is the same as selecting a help tree with option 2.

For more information about help trees, see “Wisedesk” in the *TURNOVER® Lifecycle Manager User Guide*.

### **CHGPWL: Change programmer worklist**

Change a programmer’s worklist.

For more information, see the “Programmer Worklist Manager” in the *TURNOVER® Lifecycle Manager Developer’s Guide*.

### **CHGPWLITM: Change programmer worklist item**

Change an item (object or source entry) on the Programmer’s Worklist.

For more information, see the “Programmer Worklist Manager” in the *TURNOVER® Lifecycle Manager Developer’s Guide*.

### **CHGPWLPGMR: Change worklist programmer**

Reassign the programmer associated with a given worklist. Depending on the parameters you specify, TURNOVER® Lifecycle Manager might make changes to the Worklist programmer and the Worklist item programmer for all lines on the worklist and checkout records.

### **CHGTOLIBL: Change TURNOVER® Lifecycle Manager liblist**

Edit and change library list object names.

### **CHKAPPDFN: Check application definition**

Verify objects in an application's libraries.

See "Reports" in the *TURNOVER® Lifecycle Manager User Guide*.

### **CHKCHGOBJ: Check changed object**

Check for objects that are currently checked out to one programmer (usually the person who runs it) which were changed by another programmer during to the time the first programmer had the objects checked out.

For more information, see "Using the multiple checkout feature" in the *'Emergency changes' on page 158*.

### **CHKINMBR: Check-in by member**

This command provides an alternate path to the Check in Source Members panel.

See "Checking source in and out" in the *TURNOVER® Lifecycle Manager User Guide*.

### **CHKINOBJ: Check-in by object**

This command provides an alternate path to the Check in Source for Production Object panel.

See "Checking source in and out" in the *TURNOVER® Lifecycle Manager User Guide*.

### **CHKOUTDFU: Check out DFU**

Check out S/36 DFU types.

See "Checking source in and out" in the *TURNOVER® Lifecycle Manager User Guide*.

### **CHKOUTMEN: Check out S/36 menu**

Check out S/36 menu types.

See "Checking source in and out" in the *TURNOVER® Lifecycle Manager User Guide*.

### **CHKOUTMSG: Check out message ID**

Check out one or more message IDs from a message file. This command, which you can enter on a command line, is used for checking out messages in PWM.

See *'Handling message IDs' on page 337*.

### **CMPSRCMBR: Compare source members**

Compares two source members. Use this command to produce a report or an outfile containing the differences.

### **CPYHLPTREE: Copy Wisedesk tree**

Copy a help tree and all of its categories to create a new tree.

For more information about help trees, see "Wisedesk" in the *TURNOVER® Lifecycle Manager User Guide*.

### **CPYPWL: Copy programmers worklist**

Copy a Programmer Worklist to a new worklist.

For more information, see the "Programmer Worklist Manager" in the *TURNOVER® Lifecycle Manager Developer's Guide*.

### **CPYPWLITM: Copy programmer worklist item**

Copy an item from one worklist to another.

For more information, see the "Programmer Worklist Manager" in the *TURNOVER® Lifecycle Manager Developer's Guide*.

### **CPYTOLLO: Copy TURNOVER® Lifecycle Manager library list**

Copy a library list object to create a new library list object.

### **CPYTSTDTA: Copy test data to production file**

Use this command only as a post-run command on a TURNOVER® Lifecycle Manager form to copy data from a test file to a production file. This command does *not* distribute data. (To do that, set the parameter on the TURNOVER® Lifecycle Manager form line, **Copy file FMTOPT**, to \*FROMFILE.)

### **CRTAPPDFN: Create application definition**

Create application definitions based on models provided with your system.

See the *TURNOVER® Lifecycle Manager Administrator's Guide*.

### **CRTDEMOENV: Create demo environment**

Create demonstration/training environments for a TURNOVER® Lifecycle Manager Training Workshop. This command requires additional libraries from UNICOM Systems, Inc. and can be used only by permission.

### **CRTFTFIL: Create FASTTASK files on production computer**

Set up DDM files so you can enter problems and requests into your production computer's project system that point back to project files on your development computer.

### **CRTPWL: Create a programmer worklist**

Create a Programmer Worklist. You can also do so by selecting option 20 on the Work with Project panel and F6 on Work with Programmer Worklists.

### **CRTRCVFORMI: Create recovery form (interactively)**

Interactively build a recovery form to undo changes that were installed by a previously run form. When you run the recovery form, it reverses the changes made by the first run.

### **CUSTFKEY: Set custom Fkey (F15) on PWM panel**

Customize the F15 key on the Programmer Worklist Manager panel. For example, you might use it to give quick access to the Pathfinder or Abstract/Probe system.

### **DISTDATA: Distribute data**

Use this command as a post-run command on a TURNOVER® Lifecycle Manager form to distribute data with a physical file (PF) to a remote system. TURNOVER® Lifecycle Manager will distribute data for the physical file you specify for the **File** parameter.

### **DLTCATSTR: Delete category search string**

Delete a search string in a category's task selection criteria.

For more information about help trees, see "Wisedesk" in the *TURNOVER® Lifecycle Manager User Guide*.

### **DLTHLPCAT: Delete help tree category**

Delete a help tree category.

For more information about help trees, see "Wisedesk" in the *TURNOVER® Lifecycle Manager User Guide*.

### **DLTHLPTREE: Delete help tree**

Delete a help tree.

For more information about help trees, see "Wisedesk" in the *TURNOVER® Lifecycle Manager User Guide*.

### **DLTPWL: Delete programmer's worklist**

Delete a programmer's worklist.

For more information, see the “Programmer Worklist Manager” in the *TURNOVER® Lifecycle Manager Developer's Guide*.

### **DSPCHKOUT: Display checked out members**

Display the View Checked Out Members panel.

See “Checking source in and out” in the *TURNOVER® Lifecycle Manager User Guide*.

### **DSPCOHST: Display checkout history**

Display all checkout and check-in history for any object or source member.

### **DSPFRMROL: Display rollover notification**

Display the form number threshold and message frequency information that's contained in the CHKFRMNUM data area. For more information, see the online Help for both this command and the SETFRMROL command.

### **DSPRELTO: Display current release information**

Display the current release of TURNOVER® Lifecycle Manager on your system.

### **DSTFOR: Distribute forms**

Alternate path is Distribution Menu option 2.

### **DUPREFOBJ: Create duplicate reference objects in production library**

Copy reference objects from library SOFTTURN to a library you specify. TURNOVER® Lifecycle Manager changes the object owner to a user profile name you specify.

For more information, see *'Managing object types, schemes and data' on page 86*.

### **DUPTSTOBJ: Duplicate test objects**

Duplicate test objects delivered with TURNOVER® Lifecycle Manager from the library SAVETEST to DEVTEST, ACPTTEST, and PRODTEST demonstration test libraries.

### **FASTTASK: FASTTASK project entry command**

Access the FASTTASK Helpdesk problem/request entry function.

See “Working with projects and tasks” in the *TURNOVER® Lifecycle Manager User Guide*.

### **HELPDESK: Enter TURNOVER® Lifecycle Manager helpdesk function**

Display the Helpdesk requester database.

### **LISTAUTH: List authorized TURNOVER® Lifecycle Manager users**

See “Reports” in the *TURNOVER® Lifecycle Manager User Guide*.

### **LISTDFTS: List application definitions**

See "Reports" in the *TURNOVER® Lifecycle Manager User Guide*.

### **LISTFORMS: List TURNOVER® Lifecycle Manager forms**

See "Reports" in the *TURNOVER® Lifecycle Manager User Guide*.

### **LISTPGMRS: List TURNOVER® Lifecycle Manager programmers**

See "Reports" in the *TURNOVER® Lifecycle Manager User Guide*.

### **MOVPWLITM: Move programmer worklist item**

Move a Programmer's Worklist item from one worklist to another.

For more information, see the "Programmer Worklist Manager" in the *TURNOVER® Lifecycle Manager Developer's Guide*.

### **MRGSRMBR: Merge source member**

Merges three or more source members into one new member.

### **PRGFOR: Purge TURNOVER® Lifecycle Manager forms**

Purge history records from TURNOVER® Lifecycle Manager files. Use caution to avoid purging important change history.

### **PRJCSTRPT: Project cost report**

See "Reports" in the *TURNOVER® Lifecycle Manager User Guide*.

### **PRJPTYRPT: Project priority report**

See "Reports" in the *TURNOVER® Lifecycle Manager User Guide*.

### **PRJRSRTIM: Project resource time report**

See "Reports" in the *TURNOVER® Lifecycle Manager User Guide*.

### **PRJTSKRPT: Project task report**

See "Reports" in the *TURNOVER® Lifecycle Manager User Guide*.

### **PRTHLPCAT: Print help tree category**

Print a report showing the category's selection criteria or task list.

For more information about help trees, see "Wisedesk" in the *TURNOVER® Lifecycle Manager User Guide*.

### **PRTHLPTREE: Print help tree**

Print a listing showing all the categories and subcategories defined for a specific help tree, as well as the task lists associated with them. This is the same as selecting a help tree with option 6 on the Work with Help Trees panel.

### **RCVFOR: Receive forms**

No parameters.

### **REFACTRPT: Reference activity report**

This command produces a report of all source checkout and object promotion activity related to a particular reference Project or tasks.

### **RESETFOR: Reset TURNOVER® Lifecycle Manager form**

Reset the form status from `RECOVER` to `RECVD` for a form that was distributed from another computer but did not run successfully.

### **RGZTOFILES: Reorganize TURNOVER® Lifecycle Manager files**

Submits a reorganize file job to batch. Be sure that no one is working with TURNOVER® Lifecycle Manager and that no TURNOVER® Lifecycle Manager forms or distributions are running.

### **RNMAPP: Rename an application**

Rename an application. It updates all of the associated files and history to the new application code you provide.

### **RPTFORMDST: Report form distribution**

See "Reports" in the *TURNOVER® Lifecycle Manager User Guide*.

### **RPTOBJDST: Report object distribution**

See "Reports" in the *TURNOVER® Lifecycle Manager User Guide*.

### **RPTOBJHST: Report object history**

Produce a report of object change history, by application, by object or for a particular Project reference.

See "Reports" in the *TURNOVER® Lifecycle Manager User Guide*.

### **RPTPGMRHST: Report programmer history**

Produce a report of object change history, by programmer, by object or for a particular Project reference.

See "Reports" in the *TURNOVER® Lifecycle Manager User Guide*.



### **RPTSRCUSE: Report source checkout history**

Produce a report of source checkout history, by application, by object or for a particular Project reference.

See “Reports” in the *TURNOVER® Lifecycle Manager User Guide*.

### **RSTARCSRC: Restore archived source**

Restores a member from the offline source archive. You can get the parameter information you need (Archive alias, Tape volume) from option 8 on the Work with Object History panel.

For more information, see “Working with object history” in the *TURNOVER® Lifecycle Manager User Guide*.

### **SAVARCO: Save archive objects**

Saves archived objects offline.

### **SAVARSBM: Submit object archive save**

Save objects offline that have been archived after a form run (this includes both Tform#D libraries and archived explode libraries), and then delete the archives after the save. This command runs in batch.

### **SETFRMROL: Set form rollover notification**

Set the form number threshold, and the message frequency, for notifying TURNOVER® Lifecycle Manager users that their system form numbering will shortly return to 1. These values are stored in the CHKFRMNUM data area.

### **SETGLOBE: Set global parameters**

Set TURNOVER® Lifecycle Manager system defaults.

For information about the parameters, see “Utility menu” in the *TURNOVER® Lifecycle Manager User Guide*.

### **STRPWM: Start programmer worklist manager**

Starts the Programmer Worklist Manager.

For more information, see the “Programmer Worklist Manager” in the *TURNOVER® Lifecycle Manager Developer's Guide*.

### **STRSYN: Start Synchronizer**

Starts Synchronizer.

### **TADDNEWS: Add news item**

Post news items to the TURNOVER® Helpdesk Welcome page. You can use this command in an escalation table to have TURNOVER® Lifecycle Manager automatically post a notice when an event (such as a change in status) occurs for a task.

For more information, see the *TURNOVER® Helpdesk Installation Guide*.

### **TADJPWLITM: Adjust worklist item**

Update existing worklists after an application definition's structure its library names.

### **TARAFORM: Add related applications forms**

Create related application forms once the base form has been created.

### **TCFMFORM: Confirm form**

Sends a confirmation message for a TURNOVER® Lifecycle Manager form to the TURCRARE autostart job. You might use this command on a command line or within an exit program.

### **TCHGCMPTH: Change CM method**

Change the name of the program, the name of its library, or the names of both the program and library you want TURNOVER® Lifecycle Manager to call when executing a change management method (such as moving an object, copying an object, deleting an object, and so forth) for an object.

For more information, see *'Managing object types, schemes and data' on page 86*.

### **TCHGCMSCHE: Change CM scheme**

Change a change management scheme.

For more information, see *'Managing object types, schemes and data' on page 86*.

### **TCHGPGMDTA: Change Program Data**

TURNOVER's change program data command.

### **TCHKOUTIFS: Check out IFS object**

Check out IFS objects.

For more information, see *Managing Integrated File System (IFS) objects using TURNOVER® Lifecycle Manager*.

### **TCPYCMSCH: Copy change management scheme**

Create a new change management scheme by copying an existing scheme.

For more information, see *'Managing object types, schemes and data' on page 86*.

### **TCPYPRJ: Copy project**

Copy a project entry to create a new project entry. Associated tasks are not copied and links are not created between tasks. Do not use this command to rename a project.

### **TCPYTSK: Copy task or subtask**

Copy a task or subtask from one project to another. By default, this command creates a link between the original task and the new, copied task.

### **TCRTCMSCH: Create change management scheme**

Create a change management scheme.

For more information, see *'Managing object types, schemes and data' on page 86*.

### **TCRTSRVPGM: Create Service Program**

TURNOVER's front-end to the CRTSRVPGM command.

### **TCVTOBJTYP: Convert object type**

Convert an object from one type code to another: to convert RPG to RPGLE, or RPG to RPGMOD.

For more information, see *'Managing ILE programs' on page 345*.

### **TDLTAPPDFN: Delete application definition**

Delete an application definition. TURNOVER® Lifecycle Manager deletes the application definition after it deletes all existing levels for the application. For more information, see *Chapter 2: Working with Application Definitions* in the TURNOVER® Lifecycle Manager User Guide.

### **TDLTAPPLEV: Delete application level**

Delete a level from an application definition. For more information, see *Chapter 2: Working with Application Definitions* in the TURNOVER® Lifecycle Manager User Guide.

### **TDLTCMSCH: Delete a change management scheme**

Delete a change management scheme. For more information, see *'Managing object types, schemes and data' on page 86*.

### **TDSPOBJAUD: Display object audit info**

View information about an object as it is stored in TURNOVER® Lifecycle Manager's audit database.

### **TDSPOBJD: Display object description**

View version information that TURNOVER® Lifecycle Manager stores in an object's IBM i object description. For more information, see *'Managing multiple versions' on page 161*.

### **TIMERPT: Project timesheet report**

Produce a report of source checkout history, by application, by object or for a particular Project reference.

See “Reports” in the *TURNOVER® Lifecycle Manager User Guide*.

### **TIMESHEET: Enter timesheet**

Display the daily timesheet panel. See “Working with projects and tasks” in the *TURNOVER® Lifecycle Manager User Guide*.

### **TNEWRELOBJ: New related object rule**

Change the rule that controls when and how new objects are created within a related application. For more about related applications, see *‘Managing multiple versions’ on page 161*.

### **TOPACK: Pack source member**

Pack archive source members. You can also use it and the `TOUNPACK` command to pack (compress) and unpack (decompress) your own source files.

### **TOPACKF: Pack a source file**

Packs all the members of a source file. This command should be submitted to run in batch. For more information, see the `TOPACK` command.

### **TOUNPACK: Unpack source member**

Unpacks a source member that was packed (compressed) with the `TOPACK` command.

### **TPKGFORM: Package form**

Create a save file that *TURNOVER® Lifecycle Manager* can use to distribute a form. For more information, contact your support representative.

### **TPRTTSK: Print task**

Prints a task.

### **TRGZARCF: Reorganize source archive file**

Maintain archived source (versions of source older than the current version of source) on line and off line. For more information, see “Utility menu” in the *TURNOVER® Lifecycle Manager User Guide*.

### **TRMTCFGWIZ: Run the Remote Configuration wizard**

Run the Remote Configuration wizard to send application or system definitions to remote systems. For more information, see “Remote Configuration Wizard” in the *TURNOVER® Lifecycle Manager User Guide*.

### **TRPTCON: Report conflicts**

Produce a report of related application conflicts, including an audit history of conflict resolutions. For more information, see *'Managing multiple versions' on page 161*.

### **TRSTCMSCH: Restore change management scheme**

Restore a change management scheme definition that was previously saved, and optionally restore its method programs and resource library (containing any additional objects required by the method programs), from a save file. (The save file must have been created with the TSAVCMSCH command.) For more information, see *'Managing object types, schemes and data' on page 86*.

### **TRTNOBJDTA: Set RTNOBJDTA parameter**

Use this command in a change management method program (CL program only) to parse the return data. For more information, see *'Managing object types, schemes and data' on page 86*.

### **TRTVCMPARM: Retrieve CM method parameter**

Use this command in a change management method program (CL program only) to parse the incoming **Object data** parameter into program variables. For more information, see *'Managing object types, schemes and data' on page 86*.

### **TRTVOBJDTA: Retrieve CM Object Data**

Retrieve change management object data.

### **TRTVSRCD: Retrieve stream file details**

Use this command to retrieve the stream file details. It performs like RTVMBRD for native source.

### **TRTVSRCPTH: Retrieve stream file path**

Use this command to retrieve the stream file path. This command is based on application/release/version/type code/identifier.

### **TRUNCMMTH: Run CM method**

Execute a CM method from within the method program for another CM method. For more information, see *'Managing object types, schemes and data' on page 86*.

### **TSAVARCF: Save source archive file**

Maintain archived source (versions of source older than the current version of source) online and offline.

In the application definition, you can indicate the qualified name of the source file to which source is to be archived. You can then run this command to save source offline, and update the TURNOVER® Lifecycle Manager archive history to keep track of which archived versions have been saved to which media volumes. For more information, see “Utility menu” in the *TURNOVER® Lifecycle Manager User Guide*.

### **TSAVCMSCCH: Save change management scheme**

Save a change management scheme definition, and optionally its method programs and resource library, to a save file. (The resource library contains any additional objects required by the method programs.) For more information, see *‘Managing object types, schemes and data’ on page 86*.

### **TSAVIFSARC: Save IFS archive directory**

Save an IFS archive subdirectory and, optionally, to clear the subdirectory after it is saved. For more information, see “Saving an IFS Archive Directory” in the *TURNOVER® Lifecycle Manager User Guide*.

### **TSETAPPOBJ: Set application object**

Set the initial version number for all objects in an application to a value you specify. For more information, see *‘Managing multiple versions’ on page 161*.

### **TSETOBJVER: Set object version**

Set the initial version information for the objects in an application, processing objects by name, library, and type. (See also TSETAPPOBJ above.) For more information, see *‘Managing multiple versions’ on page 161*.

### **TSETRSTDLY: Set restore delay defaults**

Set retry and delay times for the TURNOVER® Lifecycle Manager form receive process, when a form is received on a remote computer.

### **TSTRESCQ: Start escalation queue monitoring**

Starts the escalation queue monitor or changes the monitoring time interval.

This command must be submitted using the SBMJOB command; it cannot be run interactively. For more information, see *‘Autostart jobs’ on page 139*.

### **TUNPKGFORM: Unpackage form**

Receive and, optionally, run a TURNOVER® Lifecycle Manager form from a save file that was created using the TPKGFORM command.

## **TUPDHLPTSK: Update related helpdesk tasks**

Update the Helpdesk tasks associated with a Working task. This command is triggered when the Working task assumes a status of ASSIGNED or DONE.

For more information, see “Working with escalation table commands” in the *TURNOVER® Lifecycle Manager User Guide*.

## **TURNOVER: TURNOVER® Lifecycle Manager main menu**

Access the TURNOVER® Lifecycle Manager main menu or an option it from a command line. See *‘Main menu options’ on page 409*.

## **TVFYOBJAUD: Verify object audit information**

This command is for use in an exit or other program to confirm that an object’s creation information matches that of its source. Your program should monitor for abnormal termination messages.

## **TWRKCMSCH: Work with change management schemes**

View all change management schemes that have been defined to TURNOVER® Lifecycle Manager. For more information, see *‘Managing object types, schemes and data’ on page 86*.

## **TWRKCON: Work with conflicts**

View a list of related application conflict changes that have been made. For more information, see *‘Managing multiple versions’ on page 161* and the online Help.

## **TWRKLIC: Work with licenses**

View the license and PTF level of each TURNOVER® product.

## **TWRKTSK: Work with tasks**

Displays the Work with Projects and Work with Tasks panels. See “Working with projects and tasks” in the *TURNOVER® Lifecycle Manager User Guide*.

## **VIEWDFTS: View defaults**

View application definitions. See “Working with application definitions” in the *TURNOVER® Lifecycle Manager User Guide*.

## **VIEWGLOBE: View global defaults**

Displays the View Global TURNOVER® Lifecycle Manager Defaults panel.

## **VIEWPTFHST: View PTF history**

Display the history of TURNOVER® Lifecycle Manager system program temporary fixes applied to your system.

### **WDDYNUPD: Wisedesk dynamic update**

Control dynamic maintenance of help trees on your system. This command creates a data queue (if none exists) and starts, stops, or delays a job that monitors the data queue.

For more information about help trees, see “Wisedesk” in the *TURNOVER® Lifecycle Manager User Guide*.

### **WORKTASK: Work with entered problems/requests**

Access the Problem/Request review panels, normally used by a project coordinator. For more information, see “Working with projects and tasks” in the *TURNOVER® Lifecycle Manager User Guide*.

### **WRKAPPDFN: Work with Application definition**

Work with application definitions. By entering an option, you can go directly into a specific application definition panel.

See “Working with application definitions” in the *TURNOVER® Lifecycle Manager User Guide*.

### **WRKAPPGRP: Work with approval groups**

View and maintain approval groups.

For more information, see “Working with application definitions” in the *TURNOVER® Lifecycle Manager User Guide*.

### **WRKAPPLST: Work with approval lists**

View and maintain approval lists.

For more information, see “Working with application definitions” in the *TURNOVER® Lifecycle Manager User Guide*.

### **WRKCATSTR: Work with strings for category**

Work with search strings for a help tree category.

For more information about help trees and search strings, see “Wisedesk” in the *TURNOVER® Lifecycle Manager User Guide*.

### **WRKCATTSK: Work with tasks for category**

Work with the tasks associated with a help tree category.

### **WRKDSTSYS: Work with system definitions**

Displays the Maintain System Definitions panel. See “Distributing changes to production computers” in the *TURNOVER® Lifecycle Manager User Guide*.



### **WRKESQ: Work with escalation queue**

For more information, see “Working with projects and tasks” in the *TURNOVER® Lifecycle Manager User Guide*.

### **WRKFORMLOG: Work with form log**

For more information, see “Distributing changes to production computers” in the *TURNOVER® Lifecycle Manager User Guide*.

### **WRKGRPFOR: Work with form group forms**

Specify a form group, and work with all forms in that form group or to specify a form number and work with all forms that are in the same form group as that form. For more information, see “Submitting a TURNOVER® Lifecycle Manager form to run” in the *TURNOVER® Lifecycle Manager User Guide*.

### **WRKHLPTR: Work with help trees**

For more information about help trees, see “Wisedesk” in the *TURNOVER® Lifecycle Manager User Guide*.

### **WRKOBJREF: Work with object references (X-ref)**

This command is available as option 33 on the Checkout/Check-in Menu, and as option 15 on the Programmer Worklist.

### **WRKPWMDFT: Global PWM session defaults**

Displays the Global Worklist Session Defaults panel. For more information, see “Utility menu” in the *TURNOVER® Lifecycle Manager User Guide*.

### **WRKRVCJOB: Work with receive form jobs**

Find all jobs (RCVTONETF or TOAUTORCV) associated with receiving a distributed NormalKeepWithNext form.

If a form has been redistributed, you will see more than one RCVTONETF and TOAUTORCV job for the form.

### **WRKRMTAPP: Work with remote applications**

Displays the Work with Remote Application Definitions panel.

For more information, see “Distributing changes to production computers” in the *TURNOVER® Lifecycle Manager User Guide*.

### **WRKTODATA: Work with \*DATA object types**

Displays the Work with \*DATA Files panel.

For more information, see “Working with \*DATA definitions” in the *TURNOVER® Lifecycle Manager User Guide*.

### **WRKTOENV: Work with TURNOVER® Lifecycle Manager environments**

Access the TURNOVER® Lifecycle Manager environment control file. For more about TURNOVER® Lifecycle Manager environments, see “Chapter 11: Set Up Additional TURNOVER® Environments” in the *TURNOVER® Lifecycle Manager User Guide*.

### **WRKTOEXIT: Work with TURNOVER® Lifecycle Manager exit programs**

For more information, see “Utility menu” in the *TURNOVER® Lifecycle Manager User Guide*.

### **WRKTOLLO: Work with library list objects**

Displays the Work with TURNOVER® Lifecycle Manager Library List Objects panel. See “Library lists” in the *TURNOVER® Lifecycle Manager User Guide*.

### **WRKTOTYP: Work with TURNOVER® Lifecycle Manager object type file**

Displays the Work with Object Type File panel (Utility Menu option 4).

### **WRKTSKFOR: Work with task forms**

Displays the Work with TURNOVER® Lifecycle Manager Forms for Task panel. See “Working with forms” in the *TURNOVER® Lifecycle Manager User Guide*.

### **WRKUSRREF: Work with user references**

Displays the Work with User Defined References panel where you can add your own references to a cross-reference table.

## D Files and data areas

This section provides a list of important TURNOVER® Lifecycle Manager files and data areas.

---

### About TURNOVER® Lifecycle Manager files

If you are upgrading from an earlier release and you have used some of these files in programs you have written, you might have to revise your programs.

When you upgrade from a previous release of TURNOVER® Lifecycle Manager, the installation conversion procedure enables you to convert the defaults and project information into a set of test libraries. This enables you to check the setup under the new release while still using the old release for daily activity. When you are ready to commit to the new release, you reapply the upgrade to your daily libraries, converting all data to go live.

| File name | Description                         |
|-----------|-------------------------------------|
| ARCHIVE   | Source Archive                      |
| ARCHIVE2  | Offline Archive                     |
| CHKAPPF   | Check Application Definition Report |
| QAUOOPT   | PDM User-defined Option             |
| TACTIONF  | Object Action                       |
| TAGOF     | Approval Group Objects              |
| TAGOITMF  | Approval Group Object Items         |
| TALOF     | Approval List Objects               |
| TAPPAUTF  | Application Authority               |
| TAPPDFNF  | Application Definition              |
| TAPPGRPF  | Application Group Approval List     |
| TAPPLF    | Application                         |

| File name | Description                                                  |
|-----------|--------------------------------------------------------------|
| TAPPLSTF  | Application Approval List                                    |
| TAPPMSGF  | Form Completion Message                                      |
| TAPPNAMF  | Approval List Object Item                                    |
| TAPPNMGF  | Approval List Object Group Item                              |
| TAPPPGMF  | Application Notify Programmer                                |
| TAPPPTH   | Source stream file path and folder names keyed on identifier |
| TAPPRELF  | Application Relationships                                    |
| TAPPROVF  | Form Approval List                                           |
| TAPPRSFF  | Apply RSF Command Strings                                    |
| TAPPUSGF  | Approval Group Object Group Users                            |
| TAPPUSRF  | Approval List Object Entries                                 |
| TAPSWRKF  | Remote Config Wizard Application Selection                   |
| TARCUSG   | Archive File Usage                                           |
| TAUDHSTF  | Audit History Report                                         |
| TBATBLDF  | Reference Form Build in Batch                                |
| TBATMANF  | Worklist Batch Management                                    |
| TBUILDF   | Form Build File Description                                  |
| TCATAUDF  | Help Tree Category Number                                    |
| TCHKAPPF  | Check Application Definition                                 |
| TCHKBATF  | Check Batch Checkout Control                                 |
| TCHKITMF  | Check Worklist Item                                          |
| TCHKLSTF  | Check Worklist                                               |
| TCMPARMS  | Change Management Parameter Template                         |
| TCODOMTF  | Change Object Description Omit Objects                       |
| TCOMMANF  | Form (Pre/Post-Run) Command                                  |

| File name | Description                               |
|-----------|-------------------------------------------|
| TCONFLTF  | Conflict Management                       |
| TCONRSNF  | Conflict Resolution Reason Code           |
| TCONTRNF  | Conflict Transaction                      |
| TCPYFORF  | Form Copy Log                             |
| TCPYLINF  | Form Copy Line Sequencing Workfile        |
| TCPYWRKF  | Form Copy Selected Form Workfile          |
| TCREATEF  | Line Creation Command                     |
| TCVTLOGF  | Type Code Conversion Log                  |
| TCVTTYPEF | Type Code Conversion                      |
| TDAPPCM   | Deploy FTP application commands           |
| TDAPPSV   | Deploy FTP applications                   |
| TDATAEXF  | *DATA Object Exception                    |
| TDEPENDF  | Form Dependency                           |
| TDEPHSTF  | Form Dependency History                   |
| TDEVICEF  | System Device                             |
| TDFTF     | Application Defaults                      |
| TDFTPTH   | Application level source path identifiers |
| TDFTSRCF  | Application Type Code                     |
| TDISTF    | Application Distribution Default          |
| TDIST#F   | Last Used Distribution #                  |
| TDISTHF   | Distribution History                      |
| TDLOG     | Deploy FTP Job Log                        |
| TDLTLOGF  | Delta Log                                 |
| TDLTNXTF  | Next Delta                                |
| TDLTOBJF  | Delete Object Control                     |

| File name | Description                   |
|-----------|-------------------------------|
| TDSERVE   | Deploy FTP Servers            |
| TDSRVCM   | Deploy FTP Server Commands    |
| TDSTDFTF  | Distribution Defaults         |
| TDTYPCM   | Deploy FTP Type Code Commands |
| TEMPDSCF  | Temporary Description         |
| TEMPNAME  | Temporary Name                |
| TEXPLINF  | Form Line Explode             |
| TEXPLODF  | Application Explode List      |
| TEXPLOGF  | Export File Log               |
| TEXPORTF  | Export Inventory              |
| TFILTERF  | Saved Filter                  |
| TFIXLEVF  | Fix Level                     |
| TFKEYF    | Customized F-Key              |
| TFLFLTf   | Form Log Filters              |
| TFLROUTF  | Document Checkout             |
| TFLRTYPF  | Application Document Types    |
| TFOLDERF  | Form Document Line            |
| TFORFUNF  | Form Group Function           |
| TFORLOGF  | Form Log                      |
| TFORMF    | Form                          |
| TFORM#F   | Last Used Form#               |
| TFORMCMF  | Form Comment                  |
| TFORMLDF  | Build Form                    |
| TGROUPF   | Form Group                    |
| TGRPFORF  | Form Group Processing         |

| File name | Description                    |
|-----------|--------------------------------|
| TJRNLOGF  | Journaling Log                 |
| TKEYPRTF  | Keyword Protection             |
| TLANGF    | Language                       |
| TLFMOVEF  | LF Move Log                    |
| TLFMOV2F  | LF Explode Log                 |
| TLIBLF    | Form Library List              |
| TLIBLSTF  | Library List Object            |
| TLINECMF  | Form Line Comment              |
| TLINEF    | Form Line                      |
| TLINECM   | Form Line Comment              |
| TLINETMF  | Line Programming Time          |
| TLOCKF    | Form Lock Check                |
| TLOGF     | Form Log                       |
| TLSTPWLF  | Last Programmer Worklist       |
| TLSTVNDF  | Last Vendor File Used          |
| TMBRRPTF  | Audit History Members Database |
| TMLTCHKF  | Multiple Checkout Signoff      |
| TMSGF     | Form Completion Message        |
| TMSGIDF   | Message ID                     |
| TNETDSTF  | Network Method Override        |
| TNXTKEYF  | Form command – Next Key        |
| TOBJACTF  | Object Action                  |
| TOBJARCF  | Offline Object Archive         |
| TOBJCMPF  | Object Compare                 |
| TOBJRPTF  | Audit History Objects Database |

| File name | Description                             |
|-----------|-----------------------------------------|
| TOBJVERF  | Object Version                          |
| TOBJWRKF  | Audit History Workfile                  |
| TOFFHSTF  | Archived Source                         |
| TOPERF    | Authority                               |
| TPGMAPP   | Programmer Library Application Settings |
| TPGMRF    | Programmer Profile                      |
| TPGMUSRF  | Programmer User-Defined Text            |
| TPRJADTF  | Project Audit                           |
| TPRJAUTF  | Project Authority                       |
| TPRJCATF  | Timesheet Category                      |
| TPRJCMTF  | Requester Comment                       |
| TPRJCSSF  | Wisedesk Category Search String         |
| TPRJCTSF  | Wisedesk Category Matching Task         |
| TPRJDSCF  | Project Details                         |
| TPRJDYNF  | Project Dynamic Update                  |
| TPRJEMAF  | Email Address                           |
| TPRJEQCF  | Escalation Queue Command                |
| TPRJEQDF  | Escalation Queue Description            |
| TPRJEQMF  | Escalation Queue Message                |
| TPRJEQTF  | Escalation Queue Table                  |
| TPRJESCF  | Escalation Queue                        |
| TPRJHFPF  | Project Help Tree Fastpath              |
| TPRJHNCF  | Wisedesk Next Category ID               |
| TPRJLNKF  | Task Link                               |
| TPRJMKDF  | Project Details Mask Description        |



| File name | Description                     |
|-----------|---------------------------------|
| TPRJMSGF  | Project Message                 |
| TPRJMSKF  | Project Details Mask            |
| TPRJOUTF  | Project Outfile                 |
| TPRJMSP   | Microsoft Project File Names    |
| TPRJPRJF  | Project                         |
| TPRJREQF  | Requester                       |
| TPRJRMDF  | Project Report Mask Description |
| TPRJRMKF  | Project Report Mask             |
| TPRJRSRF  | Project Resource                |
| TPRJRTYF  | Requester Type                  |
| TPRJRUTF  | Requester User-Defined Field    |
| TPRJSTSF  | Project Status Code             |
| TPRJTCTF  | Wisedesk Category               |
| TPRJTIMF  | Project Timesheet File          |
| TPRJTREF  | Wisedesk Tree                   |
| TPRJTRSF  | Task Relationships              |
| TPRJTSKF  | Task                            |
| TPRJTYPF  | Task Type                       |
| TPRJUDFF  | Project User-defined Fields     |
| TPRJUD1F  | Valid values--Type 1            |
| TPRJUD2F  | Valid values--Type 2            |
| TPRJUD3F  | Valid values--Type 3            |
| TPRJUD4F  | Valid values--Type 4            |
| TPRJURDF  | User Report Parameter           |
| TPRJURHF  | User Report                     |

| File name  | Description                                                           |
|------------|-----------------------------------------------------------------------|
| TPRJVAPF   | Project Valid Application                                             |
| TPRJVMKF   | Project Valid Details Mask                                            |
| TPRJVRMF   | Project Valid Report Mask                                             |
| TPRJVRPF   | Valid Related Projects                                                |
| TPRJVRSF   | Valid Resources                                                       |
| TPRJVSTF   | Valid Status Codes                                                    |
| TPRJWDAF   | Wisedesk Authority                                                    |
| TPRMDISF   | PRMDISFORM Control File                                               |
| TPTHNXT    | Next available source path identifier for application/release/version |
| TPWLDFTF   | Programmer Worklist Session Defaults                                  |
| TPWLRELF   | Worklist Related Applications Workfile                                |
| TPWLXREF   | Programmer Worklist X-Ref Workfile                                    |
| TRCVDTF    | Application Receive Form Default                                      |
| TRCVJOB    | Receive Form Log                                                      |
| TRCVSLTF   | Create Recovery Form Selection                                        |
| TRECOVEF   | Delete Lib Error Log                                                  |
| TREFBATF   | Build Form in Batch Workfile                                          |
| TRESEQF    | Workfile for Resequencing Form                                        |
| TRMTDSTF   | Remote Application Distribution Log                                   |
| TRMTROLF   | Remote Rollback Control                                               |
| TRMTUPDF   | Remote Application Definition Update Log                              |
| TRQSTMPF   | Select Requesters of Related Helpdesk Project                         |
| TRSFLOGF   | RSF Log                                                               |
| TRTNOBJDTA | Return Object Description Data                                        |
| TSETXREF   | X-Ref Defaults                                                        |

| File name | Description                                                                 |
|-----------|-----------------------------------------------------------------------------|
| TSRCHSTF  | Source Version Archive                                                      |
| TSRCOUTF  | Checkout                                                                    |
| TSTSCODF  | Form Status Codes                                                           |
| TSYNCHKF  | Synchronizer Checklist                                                      |
| TSYNCLLF  | Synchronizer Library List                                                   |
| TSYNTIMF  | Synchronizer Time Estimate                                                  |
| TSYSTEMF  | System Description                                                          |
| TSYSWRKF  | Remote Config Wizard System Workfile                                        |
| TTGSWRKF  | Remote Config Wizard Target Sys Workfile                                    |
| TTSKEXT   | Extended task information to hold task default stream file source locations |
| TTYPEF    | Type Code                                                                   |
| TTYPCONF  | Type Code Conversion                                                        |
| TUSRCMMF  | Change Management Scheme Methods                                            |
| TUSRCMSF  | Change Management Scheme                                                    |
| TUSRCPYF  | User-defined Copy File Methods                                              |
| TUSRDFTF  | User-defined Type Code Parameters                                           |
| TUSRMTHF  | Change Management Method                                                    |
| TUSROPTF  | User-defined Option                                                         |
| TUSRTPF   | User-defined Object Type                                                    |
| TVCADMF   | Vendor Changes Administrator                                                |
| TVCSFLTF  | Vendor Changes Status Filter                                                |
| TVCSTSF   | Vendor Changes Status Code                                                  |
| TVNDAPPF  | Vendor Application                                                          |
| TVNDBATF  | Vendor Changes Batch Job Control                                            |
| TVNDCHGF  | Vendor Change                                                               |

| File name | Description                       |
|-----------|-----------------------------------|
| TVNDHSTF  | Vendor Change File History        |
| TWLXOPTF  | Worklist X-Ref Batch Option       |
| TXRFDATF  | X-Reference                       |
| TXRFILTF  | X-Reference Filters               |
| TXRFLG2   | X-Reference Library Build History |
| TXRFLOG   | X-Reference Table Build History   |
| TXRFSYM   | X-Reference ILE Symbols           |
| TXRFUSGF  | X-Ref Usage Code Text             |
| UFILESM   | MAC-PAC File                      |

### Miscellaneous files

The following files are in the TURNOVER® Lifecycle Manager product library, instead of its data library.

| File name | Description             |
|-----------|-------------------------|
| DELTAF    | Source Compare Delta    |
| TENVF     | Environment Definitions |
| TEXITF    | Exit Program Table      |
| TPTFHSTF  | PTF History File        |
| TUSRENVF  | User-environment File   |

## About TURNOVER® Lifecycle Manager data areas

TURNOVER® Lifecycle Manager uses data areas in a variety of ways. The data areas are in the TURNOVER® Lifecycle Manager Data library (usually SOFTTURND), except for TAPEDATE, which is in the product library (usually SOFTTURN).

- To set and change data area settings on your system, use the `CHGDTAARA` command.
- To see current data area settings on your system, use the `DSPDTAARA` command.

The following list identifies some important TURNOVER® Lifecycle Manager data areas.

| Data area  | Description                                                                                               |
|------------|-----------------------------------------------------------------------------------------------------------|
| AUTOAR     | TURNOVER® Lifecycle Manager global default: Set add/replace automatically (y/n)                           |
| BLDAP      | Rebuild Abstract/Probe cross-reference                                                                    |
| BLDHWK     | Rebuild Hawkeye cross-reference                                                                           |
| CASECMD    | TURNOVER® Lifecycle Manager Main menu option 12 (user-defined)                                            |
| CHKFRMNUM  | Form numbering thresholds and message frequencies (managed by the SETFRMROL command).                     |
| CMPPGM     | Source compare exit program for Synchronizer                                                              |
| COLVLCHK   | TURNOVER® Lifecycle Manager global default: Level checking on checkout (y/n)                              |
| COMPNAME   | TURNOVER® Lifecycle Manager global default: Company name                                                  |
| DBRNOSRC   | How to handle inserted LFs with no source                                                                 |
| DISTSAVE   | Distribution save time (*FORM or *DISTRIB)                                                                |
| DSTRANWRN  | Automatically distribute forms that have a RAN-WRN status                                                 |
| DTORULE    | Use the "Delete Test Object" rule                                                                         |
| FINALQ     | Finalization job queue                                                                                    |
| GENINFO    | MS Project interface: Start position, search index name                                                   |
| GRPCHK     | TURNOVER® Lifecycle Manager global default: Allow group checkout (y/n)                                    |
| INCLCK     | Include related perm-lock levels                                                                          |
| LODXREF    | Load cross-reference exit program                                                                         |
| MRGPGM     | Source Merge exit program for Synchronizer                                                                |
| NAMESTD    | Naming standard exit program                                                                              |
| PGMRTIMSHT | Controls whether or not option 8=Pgming Time appears on Form Maintenance (Billable time in Task Defaults) |
| PWL36LIBL  | Library list to use for Programmer Worklist opt 36                                                        |
| RCVJOB     | Auto-receive job name                                                                                     |

| Data area  | Description                                                          |
|------------|----------------------------------------------------------------------|
| RCVJOBQ    | Job queue for Auto Receives                                          |
| RMVXREF    | Remove cross-reference exit program                                  |
| RTGDTA     | Controls TOAUTORCV job's RTGDTA                                      |
| SENDPA     | Send in passive mode (for FTP distribution)                          |
| SRCDATA    | TURNOVER® Lifecycle Manager data area for substitute source naming   |
| SRVPGMXREF | Exclude *SRVPGM objects from cross-reference check                   |
| SYNDFTS    | Synchronizer global defaults                                         |
| TAPEDATE   | Last tape date (Stored in library SOFTTURN)                          |
| TDFTDAYM   | Time worked for day (in minutes)                                     |
| TDFTWEEKM  | Time worked for week (in minutes)                                    |
| TDSTJOBQ   | Distribution job queue                                               |
| TDSTOLDFOR | Distribute old forms                                                 |
| TENDJRN    | End journaling                                                       |
| TEXTPRCTYP | External stored procedure type code                                  |
| TGAEONLY   | Restrict access to apps to Graphical Application Designer (GAD) only |
| TGTRLS     | TURNOVER® Lifecycle Manager global default: Target release           |
| TIMDEPSCH  | Library used for CRTFILLIB on CRTTAATool of TIMDEPSCH                |
| TLIB#      | Maximum number of libraries allowed in a library list.               |
| TNO2100    | Retain Error Checking report format                                  |
| TOCMPSRC   | TURNOVER® Lifecycle Manager compare source parameters                |
| TOEXIT1    | Form finalization processing exit                                    |
| TOEXIT2    | Pre-distribution form exit                                           |
| TOEXIT3    | Pre-distribution library exit                                        |
| TOEXIT4    | Receive of distribution exit                                         |
| TOEXIT5    | Determining source file names exit                                   |

| Data area | Description                                         |
|-----------|-----------------------------------------------------|
| TOEXIT8   | Form initialization phase exit                      |
| TOEXIT9   | Post-checkout exit                                  |
| TOEXIT10  | Manual check-in exit                                |
| TOEXIT11  | Form completion processing exit                     |
| TOEXIT12  | Add a line to TURNOVER® Lifecycle Manager form exit |
| TOEXIT13  | Pre-run error check exit                            |
| TOEXIT14  | Pre-checkout processing exit                        |
| TOEXIT15  | Add Programmer Worklist (PWL) item exit             |
| TOEXIT16  | Form approval exit                                  |
| TOEXIT19  | Pre-form group exit                                 |
| TOEXIT20  | Post-form group exit                                |
| TOEXIT21  | Post-receive exit                                   |
| TOEXIT22  | Post add-line exit                                  |
| TOEXIT23  | Synchronizer create change impact file exit         |
| TOEXIT26  | Related forms creation exit                         |
| TOEXIT27  | Remote Configuration Wizard pre-save exit           |
| TOEXIT28  | Remote Configuration Wizard post-receive exit       |
| TOEXIT29  | Form confirmation exit                              |
| TOEXIT30  | Conflict resolution exit                            |
| TOEXIT31  | Lock check exit                                     |
| TOEXIT32  | Change checkout exit                                |
| TOEXIT33  | Application pre-checkin exit                        |
| TOEXITXN  | Create command replacement variable exit            |
| TOPROJ01  | Timesheet exit                                      |
| TOPROJ02  | Escalation blackout date/time exit                  |

| Data area    | Description                                                                    |
|--------------|--------------------------------------------------------------------------------|
| TOPROJ03     | Escalation blackout item exit                                                  |
| TOPROJ04     | Task authorization exit                                                        |
| TOPROJ05     | Task validation exit                                                           |
| TOEXITXREF01 | Cross-reference exit 1: RPG source scan                                        |
| TPROTECT     | Protect parameter keyword                                                      |
| TROLLBACK    | Rollback all systems default                                                   |
| TSVDFTS      | Distribution SAVLIB defaults                                                   |
| TSCHEMELIB   | TURNOVER® Lifecycle Manager Change Management Scheme Library                   |
| TURDS36      | TURNOVER® Lifecycle Manager global default: Manage S/36 objects                |
| TURDUNIQ     | TURNOVER® Lifecycle Manager global default: Require unique names (y/n)         |
| TURDVXRC     | TURNOVER® Lifecycle Manager view cross reference commands                      |
| TURNESCQ     | TURNOVER® Lifecycle Manager escalation queue delay interval                    |
| TURPGMR      | TURNOVER® Lifecycle Manager global default: Programmer check at check-in (y/n) |
| TURUSR       | TURNOVER® Lifecycle Manager global default: Submit user                        |
| XREFOVRFLW   | Cross-reference overflow threshold                                             |





**[www.softlanding.com](http://www.softlanding.com)**

We welcome feedback on our documentation. Please email us at:  
[tech.authors@softlanding.com](mailto:tech.authors@softlanding.com)

**[www.unicomglobal.com](http://www.unicomglobal.com)**