

TURNOVER® Lifecycle Manager

Developer's Guide

Publication information

TN4D-2100-00 (January 2021)

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 2005-2021 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 2005-2021 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.

Contents

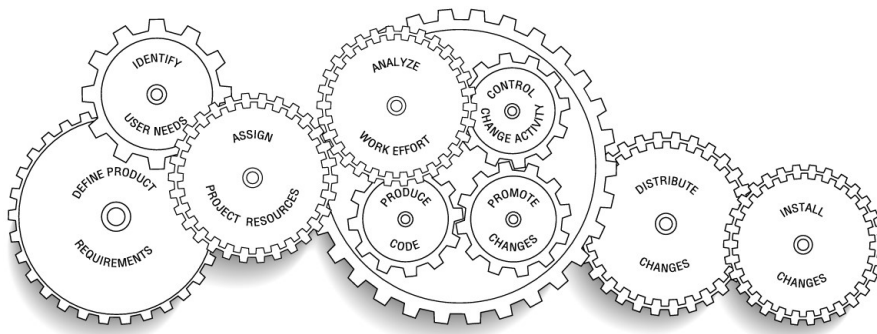
1 Overview of TURNOVER® Lifecycle Manager	5
Project-driven change management	5
Application diagrams	6
Tasks	7
Before using TURNOVER® Lifecycle Manager	10
2 Examples of change management productivity	11
Example 1: Adding a field to a PF	11
Example 2: Changing a set of objects	12
Example 3: Building a list of objects to change	13
Example 4: Copybook code	14
Example 5: Rolling out changes	15
Example 6: Changing a printer file	16
More examples	17
3 Typical change workflow	19
Entering problem reports and change requests	20
Reviewing and assign problems and requests	22
Analyzing tasks	24
Creating worklists	27
Populating worklists	29
Checking cross-references	32
Adding new items to the worklist	34
Checking out objects and source	35
Coding and testing changes	39
Compiling the objects	42
Creating TURNOVER® Lifecycle Manager forms	49
Checking forms for errors	56
Getting forms approved	57
Running forms	57
Testing and acceptance	60
Promoting to a production environment	61
Distributing to production computers	62
Updating the project task	62

Recording task time	63
4 Advanced topics	65
More on the Programmer Worklist Manager	65
Setting user-defined PWM options	66
Setting TURNOVER® Lifecycle Manager user-defined options in PDM	67
Emergency changes	68
Using the multiple checkout feature	70
Resolving concurrent changes	70
TURNOVER® Lifecycle Manager commands	71
APIs, exits, overrides	75
Cross-referencing	78
Using the source compare utility	80
Using the source merge utility	80
Managing vendor software changes	80
Application versioning	81
Object versioning	82
Message IDs, message files, menus, and other object types	82
Managing ILE programs	84

1 Overview of TURNOVER® Lifecycle Manager

Project-driven change management

TURNOVER® Lifecycle Manager has redefined change management to be a framework for software development – and redevelopment, the activity most developers do most of the time.



TURNOVER® Lifecycle Manager has many component parts that are integrated into one system. It includes:

- helpdesk and online user-entry of change requests
- project management system
- source control management
- analysis tools such as a built-in cross-referencing and version control facilities
- programmer 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
- vendor software synchronization package, including source compare and merge

- CASE & other development tool interfaces.

In the broadest sense of the phrase, change management defines the primary role of most information systems organizations. The TURNOVER® Lifecycle Manager change management system is *your* departmental application. As a developer, your role can be described as a *change agent*. Implementing something new means defining a migration path from what *is* to what *needs to be*. That is what is meant by *concurrent software redevelopment*. It describes the reality with which we all must contend. We are continually redeveloping software as it is being used. How could you hope to do that well without a system?

This manual describes change management and how the TURNOVER® Lifecycle Manager system helps you be a more effective change agent.

More information

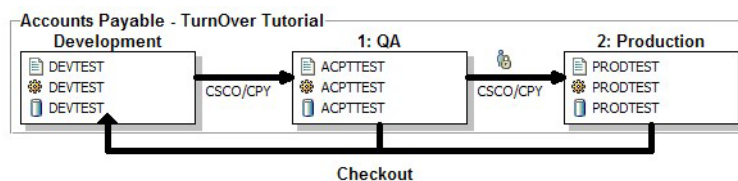
For an introduction to TURNOVER® Lifecycle Manager, see the *TURNOVER® Lifecycle Manager Administrator's Guide*.

For information about defining applications, checking out objects and source, the promotion request (TURNOVER® Lifecycle Manager form), TURNOVER® Lifecycle Manager Projects, and distribution, see the *TURNOVER® Lifecycle Manager User Guide*.

Application diagrams

The heart of any change management system is the way in which it enables you to specify defaults and rules for managing your applications. The application schematic enables you to visualize what is going on.

Sample application diagram



Each box represents a set of libraries. Development libraries are on the left; production libraries are on the right. You can have any number of intermediate libraries (ACPTTEST stands for Acceptance). You can also define vendor libraries (not shown) from which source is checked out and multiple data libraries ("explode data libraries", not shown) into which data objects can be duplicated. Checkout occurs from *right to left*; promotions proceed from *left to right*. The development libraries can be a common development library as is implied by the diagram, or individual programmer libraries. Alternatively, you can have project libraries (libraries used for development for the duration of a project), if you want.

In the diagram, each pair of “from” and “to” libraries is defined in an *application definition* and is referred to as a *level* of the application. In our schematic example, DEVTEST to ACPTTEST is application AP level 1; ACPTTEST to PRODTEST is application AP level 2. Checkout rules, promotion rules, and defaults are all specified in an *application definition*. These can be different for each *level*.

Application definition rules

The application definition contains rules such as:

- Do you want to retain (archive) the old object when it is replaced by a new object?
- How many copies of source should be retained in the source archive at each level?
- Do you want to enforce naming standard rules for the application?
- What method should be employed when promoting source and objects? (CSCO, which stands for Copy Source, Compile Object, is the most common method. However, CSMO (Copy Source, Move Object; MO, Move Object; CD, Create Duplicate, and CS Copy Source (only) all have their uses).
- Do you want to distribute objects, object and source, or only source to one or more other computers at the time objects are promoted?
- Does anyone need to approve changes (on line) before they are run?
- What profile will own objects created by TURNOVER® Lifecycle Manager?
- What object types are valid for this application?

This is just a sampling of rules contained in the TURNOVER® Lifecycle Manager application definition.

Some of the answers may differ depending upon which level is being defined. For example, you may archive a minimal number of source generations when you promote to acceptance (level 1); but several generations when you promote to production (level 2). Certainly, the list of people who need to approve a change may differ at each level. And, it is more likely that you will distribute changes when they are promoted to production than to acceptance test.

More information

For more information about application definitions, see *TURNOVER® Lifecycle Manager Administrator's Guide* and the *TURNOVER® Lifecycle Manager User Guide*.

Tasks

Once you have been assigned a task and have determined what objects you need to change or create (more on how TURNOVER® Lifecycle Manager supports these activities later), you can do the following:

Check out objects and source from production

TURNOVER® Lifecycle Manager copies the source from your production source library to your development library. Depending upon how your application is set up, TURNOVER® Lifecycle Manager does not allow others to check out the same objects. (For emergency changes, see *'Emergency changes' on page 68.*) Non-source objects are recorded as checked out and duplicated.

Develop objects in test

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

Check cross-reference

You will want to check the TURNOVER® Lifecycle Manager cross-reference files to see if there are any related objects that need to be changed, either in the application in which you are working or in other applications.

Create a TURNOVER® Lifecycle Manager form

When you are ready to promote source and objects, you build a TURNOVER® Lifecycle Manager form. A *form* contains 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 OVRDBF to be run when objects are promoted.

Pre-run error-check form

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

Obtain form approval

Depending on whether or not your management requires on-line approval, you will notify all of the people on the approval list by pressing a command key. Each user signs on, and reviews and approves your form. You can also print a hard copy of the form and affix signatures to that, if you want.

Run form

Your Administrator or you submit the form to run. Forms can be scheduled to run at a date and time you specify.

When it runs, source is copied to the target library and objects are compiled into the appropriate target object libraries. Non-source objects can be duplicated or moved. Source that you do not want compiled, such as RPG /COPY code or S/36 OCL, is copied. Old source is archived; old objects are archived. TT (Archiving of source or objects is optional and the number of generations retained can vary depending on how you design your archive process.) Source and objects in the From library are deleted (optionally). If errors occur, your changes can be rolled back automatically.

Check out from acceptance (or any intermediate level)

If errors are detected during acceptance testing, then you will check those objects out *from acceptance test*, revise them, and then promote them for acceptance testing again. This may happen several times (except for *your* code, of course).

Copy form

Once acceptance testing is complete you can then promote objects into production. This is done by copying your original form to create a new form and then running that new form. All library references and similar defaults are translated when a form is copied. Usually, a TURNOVER® Lifecycle Manager Administrator performs this and the following steps.

Run production form

Again, the form is submitted to run. You can schedule it to run at any time. When it does run, it is rolled back if errors are detected.

Back out form

If you are not satisfied with the result of running a form, it can be rolled back. Any subset of objects can be rolled back, if necessary.

Distribute form

If you are running a network of IBM i systems, TURNOVER® Lifecycle Manager can distribute a form after it runs successfully. This can happen immediately, or at some later date and time. When a form is distributed, the objects are duplicated into a distribution library and, together with TURNOVER® Lifecycle Manager control information, written to tape or sent over your network.

Receive and run form

Forms are received and run on your remote production computers just as they were run on the development computer. If you distribute *from* AP level 2, you need a definition for AP level 2 on the remote computer. That application definition controls how the objects are managed on *that* computer. Usually only objects are distributed; therefore the method at the remote computer is MO (Move Object); but you can send source only, or source and objects, and compile objects on the target computer, if necessary.

Update form status

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

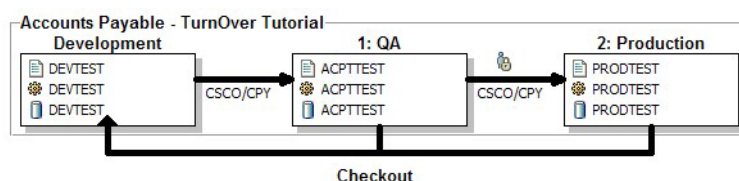
For more information about how TURNOVER® Lifecycle Manager can help you be more productive, see ['Examples of change management productivity' on page 11](#).

Before using TURNOVER® Lifecycle Manager

Obtain and study your application diagrams

The first step is to know how your applications are managed. Obtain application diagrams from your TURNOVER® Lifecycle Manager administrator. These diagrams show how your application is managed, as well as the names of libraries at each level.

This is an example of an application diagram.



Establish diagrams like this one for each application to ensure that everyone knows how you are managing them.

Are you enrolled in TURNOVER® Lifecycle Manager?

Each TURNOVER® Lifecycle Manager user must be enrolled and authorized to perform functions within TURNOVER® Lifecycle Manager. To find out what you are authorized to do and upon which applications you are authorized to work, see your TURNOVER® Lifecycle Manager administrator. If you are a developer, you also have a programmer profile that identifies your personal library.

Are you authorized to use TURNOVER® Lifecycle Manager Projects?

If your company uses TURNOVER® Lifecycle Manager Projects, each TURNOVER® Lifecycle Manager Projects user must be authorized to work with the Project system, and must also be authorized to one or more projects. With Project authority, you can view, add, change, and/or delete tasks within a designated project. To find out which projects you are authorized to use, see your TURNOVER® Lifecycle Manager administrator. The examples in this guide assume that you are using TURNOVER® Lifecycle Manager Projects.

2 Examples of change management productivity

The following examples illustrate the features of TURNOVER[®] Lifecycle Manager and provide a synopsis of how to approach a problem or perform a task.

Example 1: Adding a field to a PF

Suppose you want to add a field to a PF and recompile all of the logicals and programs that refer to it.

Without TURNOVER[®] Lifecycle Manager

Without TURNOVER[®] Lifecycle Manager, you would have to:

- 1 Research all of the program and logical file source for references to the PF.
- 2 Write a list of these objects.
- 3 Copy the source from production to a development library.
- 4 Change the source using SEU.
- 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

With TURNOVER® Lifecycle Manager, you would:

- 1 Check out the PF source.
- 2 Change the source in SEU.
- 3 Compile to new PF in development.
- 4 Build a promotion request form and add the related LFs and Programs automatically.
- 5 Submit the form.

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

You want to undo your changes after they are installed?

Without TURNOVER® Lifecycle Manager, you would have to reverse all of the former steps.

With TURNOVER® Lifecycle Manager, you just build a Recovery form and run it. The rest is automatic.

Example 2: Changing a set of objects

Suppose you want to change a set of objects that were changed months ago.

Without TURNOVER® Lifecycle Manager

Without TURNOVER® Lifecycle Manager, you would have to:

- 1 Find a written list of the objects that were changed, or display object description to a file.
- 2 Query the file to find one of the objects that was changed on or about the date in question.
- 3 Find all of the other objects changed on the same date.
- 4 Hope none had been changed since that date.
- 5 Copy the source from production to development for each related object.
- 6 Change the objects in development using SEU.
- 7 Compile and test the changes.
- 8 Copy the source in production to a backup (archive) file.
- 9 Copy the source from development to production.
- 10 Back up the production programs and related objects
- 11 Call all of the users to get them off of the system or report to work at 3 am
- 12 Recompile the programs and related objects
- 13 Set object ownership and grant object authority appropriately

14 Remember to delete the source in development (cleanup)

15 Hope you have not missed anything or made an error

With TURNOVER® Lifecycle Manager

With TURNOVER® Lifecycle Manager, you would:

- 1 Look up the original change request.
- 2 View a list of objects changed to respond to that request.
- 3 In a single step, check out all of the related objects on the list.
- 4 Change the source in development using SEU.
- 5 Compile the objects in development and test them.
- 6 In a single step, build a new promotion request form.
- 7 Submit the TURNOVER® Lifecycle Manager form to run at 3 am.

And rest assured that you have not forgotten anything.

Example 3: Building a list of objects to change

Suppose you want to build a list of objects to change and then check out everything on that list.

Without TURNOVER® Lifecycle Manager

Without TURNOVER® Lifecycle Manager, you would have to:

- 1 Research the programs affected
- 2 Write down each program and related object
- 3 Check each related object to determine whether they need to be changed, recompiled, or left unchanged.
- 4 Copy the source for each object to be changed from production libraries to development
- 5 Notify everybody else about the pending changes (and hope no one forgets)

With TURNOVER® Lifecycle Manager

With TURNOVER® Lifecycle Manager, you would:

- 1 Using a PDM-like facility called the Programmer Worklist Manager (PWM), select the objects you want to change from a list of related objects.
- 2 In a single step, check out all of the objects you want to change.

Example: Changing an IBM i menu

Suppose you want to check out and change an IBM i menu.

Without TURNOVER® Lifecycle Manager

Without TURNOVER® Lifecycle Manager, you would have to:

- 1 Copy the menu display file source to your development library.
- 2 Copy the menu message file to your development library.
- 3 Modify the menu using SDA.
- 4 Copy the production menu DDS source to backup (archive) library.
- 5 Copy the menu DDS source from your development library to a production source file.
- 6 Copy the production menu message file to a backup (archive) library.
- 7 Copy the menu message file from development to a production source file.
- 8 Copy the menu command source to production source file.
- 9 Run the create menu command to create the menu in your production library.
- 10 Set object ownership and grant object authority appropriately.

With TURNOVER® Lifecycle Manager

With TURNOVER® Lifecycle Manager, you would:

- 1 Check out a *MENU object type using a PDM-like command option.
- 2 Modify the menu using SDA.
- 3 Include the menu entry on a promotion request form (all menu components are handled automatically).
- 4 Submit the TURNOVER® Lifecycle Manager form.

TURNOVER® Lifecycle Manager handles the details you might have overlooked.

Example 4: Copybook code

Suppose you want to recompile program CEM0032 every time /COPY copybook code CEMABC is modified, and always want to compile program CEM0032 as User profile *OWNER? (It is critical that this be done consistently every time, no matter who is assigned the task.)

Without TURNOVER® Lifecycle Manager

Without TURNOVER® Lifecycle Manager, you would have to:

- 1 Maintain a list of programs that contain reference to copy code.
- 2 Maintain a list of programs that must be compiled with User profile *OWNER (as well as list of other create command default exceptions).
- 3 Notify the entire staff of the existence of these exception lists.
- 4 Hope that everyone remembers to refer to them and updates them, whenever necessary.
- 5 Change the source member using SEU.
- 6 Compile and test programs referring to the changed copybook code.
- 7 Recompile the program in production specifying the User profile *OWNER parameter.

With TURNOVER® Lifecycle Manager

With TURNOVER® Lifecycle Manager, you would:

- 1 Check out the copybook code using a PDM-like option.
- 2 Change the source member using SEU.
- 3 Compile and test the programs referring to the copy code.
- 4 Include the copybook code on a promotion request form (the rest is automatic).

TURNOVER® Lifecycle Manager includes all of the related objects on the form and, referring to create overrides entered the first time the object was created, creates program CEM0032 with User Profile *OWNER.

Example 5: Rolling out changes

Suppose you promoted several programs and related objects and received a call from a now irate VP of Marketing insisting that you roll out the changes immediately!

Without TURNOVER® Lifecycle Manager

Without TURNOVER® Lifecycle Manager, you would have to:

- 1 Write down all of the objects you just promoted.
- 2 Find the previous source (or load it from a backup).
- 3 Copy all of the source now in production into a development library (you do not want to lose your changes).
- 4 Copy all of the old source into production source files.
- 5 Compile the objects from the production source.

- 6 Is data involved? Hope you have been journaling or have a recent backup.
- 7 Set object ownership and grant object authority appropriately.
- 8 Hope you did not miss anything.

With TURNOVER® Lifecycle Manager

With TURNOVER® Lifecycle Manager, you would:

- 1 Create a recover form and submit it.

(Data objects require special consideration *with* TURNOVER® Lifecycle Manager, also.)

All of the objects are backed out, the previous source and objects restored, and the modified source returned to the Developer's library, automatically.

Example 6: Changing a printer file

Suppose you need to change a printer file, and want to make sure that all of the create parameters are preserved when the file is recreated in production.

Without TURNOVER® Lifecycle Manager

Without TURNOVER® Lifecycle Manager, you would have to:

- 1 Copy the printer file source to a development library.
- 2 Change the printer file using RLU (or SEU).
- 3 Compile the printer file.
- 4 Compile the programs in development from production source and test them.
- 5 Copy to production source to a backup (archive) source file.
- 6 Copy the production object to a backup library.
- 7 Copy the source from your develop library to production.
- 8 Set the create parameters just as they are on the existing printer file.
- 9 Compile the printer file in production.
- 10 Compile or recompile all of the programs that refer to the printer file.
- 11 Set object ownership and grant object authority appropriately.

With TURNOVER® Lifecycle Manager

With TURNOVER® Lifecycle Manager, you would:

- 1 Check out the printer file using a PDM-like command option.

- 2 View and check out (if appropriate) related programs from a list presented to you by TURNOVER® Lifecycle Manager.
- 3 Change the printer file using RLU (or SEU).
- 4 Compile the printer file.
- 5 Compile the programs in development from production source and test them.
- 6 Build and run a promotion request form.

Let TURNOVER® Lifecycle Manager handle the details.

More examples

Checking for an object lock

Suppose 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 pull the rug out from under active users.

Changing a PF with 8 members

Suppose you need to change a PF with 8 members, all with live data.

With TURNOVER® Lifecycle Manager, your physical files members are preserved whenever you make a change to a file.

Changing a database file format twenty times

For example, suppose 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.

Identifying changed objects

Suppose you need to identify what objects were changed when a problem was fixed in January of 1992.

With TURNOVER® Lifecycle Manager, you can easily retrieve a list of objects related to any historical task.

Verifying source and objects

For example, suppose you want to verify all of the source and objects before turning them over?

With TURNOVER® Lifecycle Manager, you can run an option to pre-check your changes, before submitting the TURNOVER® Lifecycle Manager job.

Promoting programs

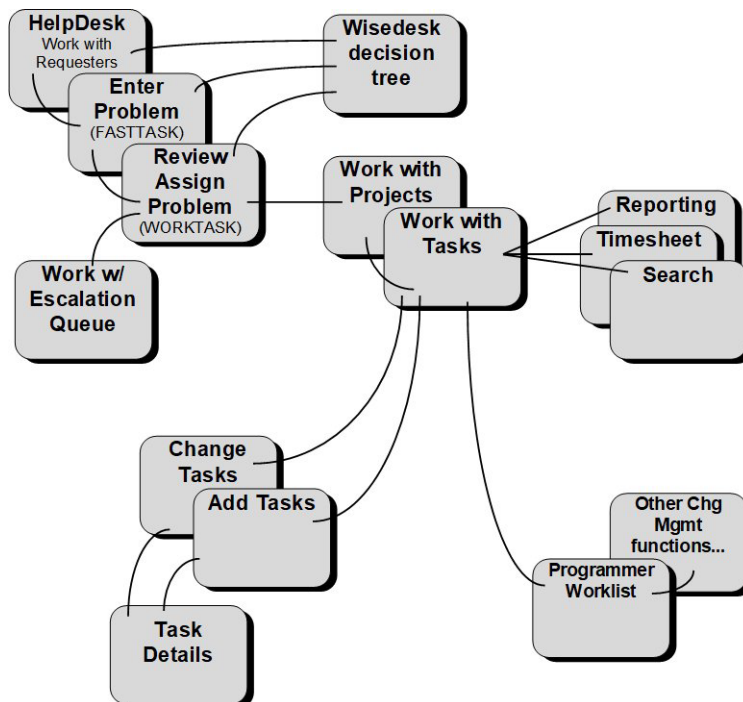
For example, suppose you want to promote 200 programs and related objects at 3 am, but want to get a good night's sleep.

With TURNOVER® Lifecycle Manager's many verification checks and automatic rollback, your likelihood of success is greater; but with TURNOVER® Lifecycle Manager, if an error occurs rollback is automatic.

3 Typical change workflow

This manual is organized chronologically around the activities normally performed by a developer. In **TURNOVER® Lifecycle Manager**, you can manage those activities using a system called **TURNOVER® Lifecycle Manager Projects**.

This map shows some of the things you can do using **TURNOVER® Lifecycle Manager Projects**.



Other tasks, such as entering a change request, are included to provide the overall context for change management and to illustrate how these tasks are done using **TURNOVER® Lifecycle Manager**. The degree to which you use these features is up to your management. For example, **TURNOVER® Lifecycle Manager** includes a Helpdesk facility; you can choose to use it, or not. Entering a problem or change request is usually performed by a user, a Helpdesk person, a manager in your department or even by you.

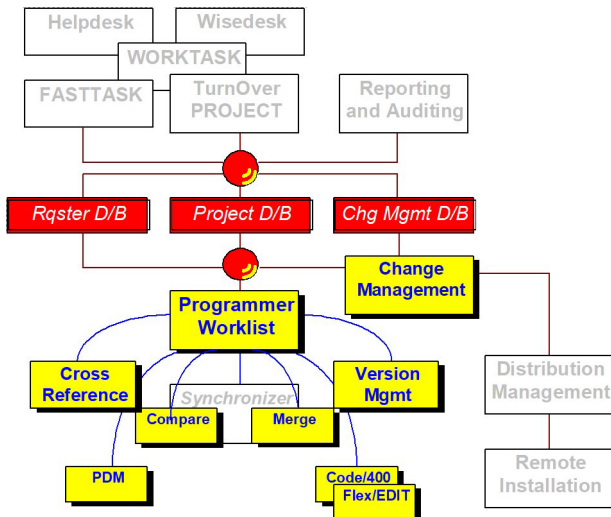
For more information about the Helpdesk and Project systems, see the *TURNOVER® Lifecycle Manager User Guide*.

Using **TURNOVER® Lifecycle Manager**

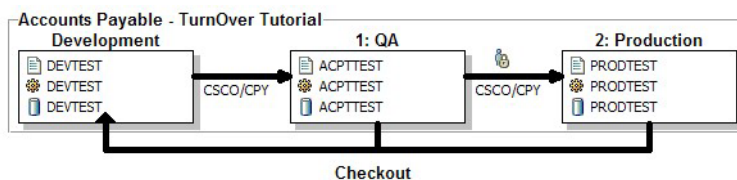
There are three methods of working with **TURNOVER® Lifecycle Manager**. You can:

- use the TURNOVER® Lifecycle Manager menus
- enter TURNOVER® Lifecycle Manager commands while working in Programmer Development Manager (PDM)
- use TURNOVER® Lifecycle Manager's own PDM-like facility, the Programmer Worklist Manager (PWM).

PWM (the "worklist") integrates all of the processes you need to perform into one panel. It provides the most efficient development path using TURNOVER® Lifecycle Manager.



This section uses a project task example to demonstrate ways you can manage work using TURNOVER® Lifecycle Manager Projects and PWM. The example is based on this application definition:



Entering problem reports and change requests

TURNOVER® Lifecycle Manager Projects enables end users and Helpdesk personnel to enter a change request or problem report directly into TURNOVER® Lifecycle Manager's project system. You can limit users to only those capabilities and projects you want them to see. Messaging features ensure that a project coordinator is informed of high priority problems as soon as they are entered. You can collect descriptive details you need to respond to requests and problems by defining text masks.

You can define a generic project like "Problems" or "Requests" (referred to as a *Helpdesk project*) to enable users enter a problem or a request and text details.

To access Work with Problems and Requests

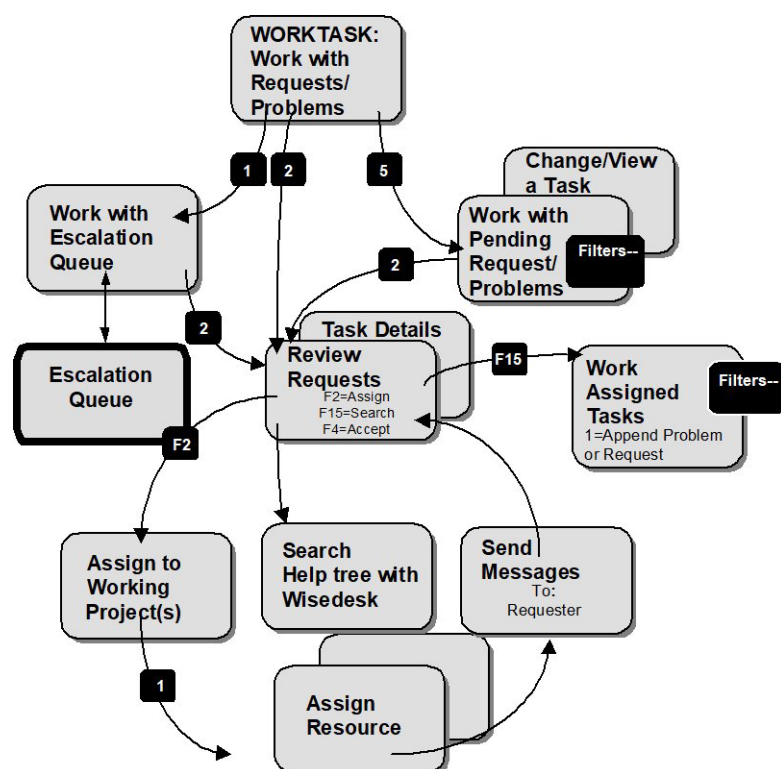
Type this command on a command line (or include it on your menu):

SOFTTURN/FASTTASK

(SOFTTURN is the default product command library, but it might be different on your system.)

Panels in FASTTASK

The panels available in FASTTASK include the following:



Entering a request

When you enter a request, the Add a Problem Report panel appears.

MM/DD/YY 10:32:05		Add a Problem Report		Your Company, Inc.	
Project: PROB All Problems and Requests					
Description <u>Change credit-term verify program</u>					
Resource <u>PSCHMIDT</u>		F4=List Phil Schmidt			
Status <u>N NEW</u>		F4=List			
Requester <u>MILLERERNE</u>		F4=List Ernest Miller			
Priority <u>1</u> 1 to 3		Sequence			
Date needed <u>0/00/00</u>		Task type . . . <u>A</u> ASAP			
Subsystem					
Pgm w/error (optional)					
F3=Exit F6=Add requester F8=User fields F12=Cancel					
F10=View 3 F11=View 2 F17=Filters F18=Timesheet F21=System command					

Up to 10 user-defined fields may be defined. "Subsystem" and "Pgm w/error" are examples.

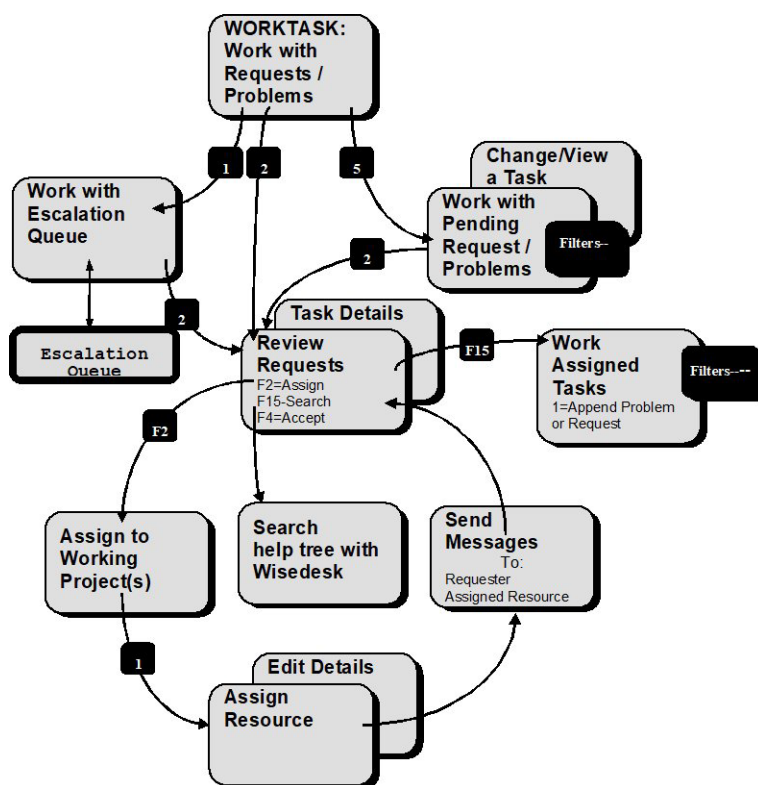
When you enter a problem or request, you might be prompted to include a more detailed description. More than one text mask may have been defined for any project. For example, if you set up a Helpdesk project for hardware problems, you might set up different text masks for "Printer Problems", "PC Problems", "Communications Problems", and so on. Select the one that is appropriate and then type the details requested. If a user is typing their own request, the masks may contain suggestions to help the user resolve a problem independently.

For more information about the Helpdesk and Project systems, see the *TURNOVER® Lifecycle Manager User Guide*.

Reviewing and assign problems and requests

When a problem report or change request is entered, the person responsible for reviewing these work requests (the project *coordinator*) reviews and evaluates them. They might do this daily or whenever a message arrives indicating that a high-priority problem or change request has been entered.

To do this, use the `SOFTTURN/WORKTASK` command.



```
MM/DD/YY      10:45:10      Work with Request      Your Company, Inc.
                        Project: PROB All Problems and Requests      YOURSYS

Position to task . . . ____ *TOP, *BOT  Apply filters . . . Y  Fl7=Filters
2=Change 3=Copy 4=Delete 5=View 6=Print 9=View time 7=Categories
10=Message 11=Move 12=Assigned task 25=Find text 1=Review

                        Assigned
Task  Description                        Task      Status
_ 0001 Daily Time Sheet won't allow you to exit after a  PRPR0021 ASSIGNED
_ 0002 Allocation will add to form regardless of status  AC310225 +ASSIGNED
_ 0003 Error adding FICA and FUCA after close           PRPR0035 DONE
_ 0004 PRM00544 submitted job can not allocate tape dri PRPR0036 ASSIGNED
_ 0005 Payroll needs reapproval if not run on the same  PRPR0037 CLOSED
_ 0006 Copy of payroll employee failed when dev not=    PRPR0049 ASSIGNED
_ 0007 When account is created no budget can be assigne AC410229 DONE
_ 0008 View Accounting Defaults - Subfile position fail  AC410231 ASSIGNED
_ 0009 Daily time sheet's F4 should allow the selection  REVIEWED
_ 0010 Add "Acct type" field to target reallocated acco  NEW
_ 0011 Fix handling of Interview Response letter        NEW
1 0012 Change credit-term verify program                 APPR0004 ASSIGNED
_ 0013 Rerun of payroll list left workfile in user libr  REJECTED  More...
F3=Exit  F5=Refresh  F6=Add Request      F2=Escalation queue F7=Review all
F10=View 3 F11=View 2 F12=Cancel F13=Repeat F18=Timesheet F21=System command
```

When a task is accepted and assigned, it is copied to a *working* project and assigned to a *resource*. The original *Helpdesk* request is updated to indicate the new project/task identifier. In this example shown above, there are several requests that are “Assigned”; some that are “Done;” and some that are “New,” which have not been reviewed yet. Your user or Helpdesk person can see the progress by viewing that original request and by reviewing the assigned task. If a task has been assigned, the status displayed is the status of the Working task. For example, your problem reports may be entered into a Helpdesk project called “Problems”. Your user does not need to

know how your working projects and personnel are organized. They might not know enough about the problem to know to which particular application it relates, nor should they. As the coordinator reviews each problem, it is assigned to specific working projects such as "Accounts payable", "Order entry", or "Inventory". (How your projects are organized depends on how your company has set up TURNOVER® Lifecycle Manager Projects.)

Analyzing tasks

TURNOVER® Lifecycle Manager enables you to build a list of objects you want to change or create. You do this by using the Programmer Worklist Manager panel and by using user-defined options you can invoke from PDM, or from tools like Hawkeye's PATHFiNDER, HelpSystems' Abstract, and TURNOVER® Lifecycle Manager's own cross-reference system.

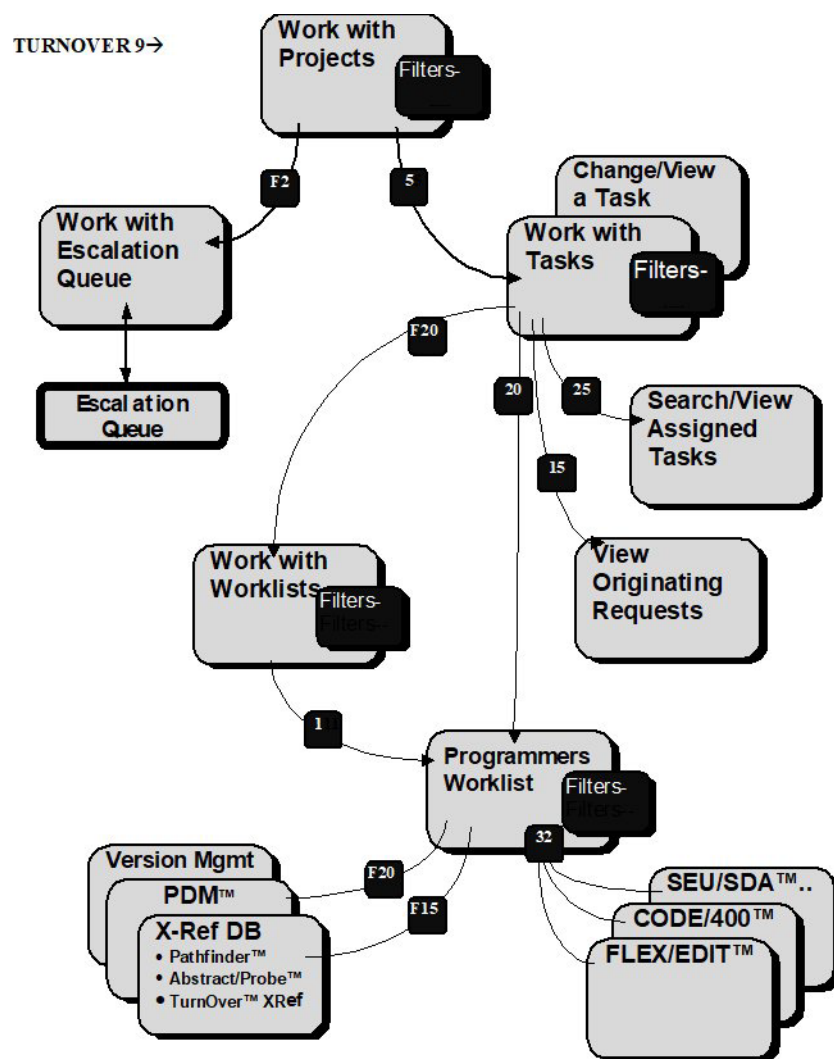
For example, assume that you have been handed a task assignment or project task report describing your assigned task.

- If this is a report, you can view and work with the task details: on the command line, type `TURNOVER`, and then select option 9 (or type `TURNOVER 9`).
- If you do not use TURNOVER® Lifecycle Manager Projects, type `SOFTTURN/STRPWM` on a command line to create a worklist. You have to make up a key as a task identifier. For more information, see your TURNOVER® Lifecycle Manager administrator.

Common steps

- 1 Find your assigned task in project task list.
- 2 Read the task details and description.
- 3 Create a Programmer worklist.
- 4 Add objects to your worklist (when viewing PDM or some other tool).
- 5 Check for related object is TURNOVER® Lifecycle Manager's cross-reference (Add items, if appropriate).
- 6 Add new object names (objects you will create) to the worklist.
- 7 Check out source for production objects (and reserve new object names).
- 8 Edit, compile and test the objects.
- 9 Create a TURNOVER® Lifecycle Manager promotion form.
- 10 Submit the promotion form to run.

TURNOVER® Lifecycle Manager projects and PWM panel map



Accessing the Work with Projects panel

- 1 On the TURNOVER® Lifecycle Manager main menu, select option 9.

The Work with Projects panel appears.

```

MM/DD/YY  10:50:15                Work with Projects                Your Company, Inc.
                                YOURSYS

Position to project . . . . ____
Filter by Administrator . . ____ Status . . . . ____ F4=List

1=Add task  2=Change  3=Copy  4=Delete  5=View  6=Print  7=Rename  9=View time
12=Work with tasks  14=Authority  16=Forms  17=Checkout  18=History

    Proj  Description                                Administrator Status
12 APPR  Accounts Payable Requests and Problem        QSECOFR    ONGOING
__ ARPR  Accounts Receivable Request and Problems    QSECOFR    ONGOING
__ HARD  All Hardware Problems and Requests          QSECOFR    ONGOING
__ TPR3  TurnOver Release 3 Problem Reports           QSECOFR    CLOSED
__ TP31  TurnOver Program Fixes to Release 3.1        QSECOFR    ONGOING
__ TP32  TurnOver Release 3.2 Problem Fixes           MARKPHIP   ONGOING
__ TP40  TurnOver Release 4.0 Problems                MARKPHIP   ONGOING
__ TQA   TurnOver Q&A Database                        MARKPHIP   ONGOING
__ TSYN  Turnover/Synon Interface                    ELMSMIKE   ONGOING
__ TXM   TurnOver Exit Modifications                 MARKPHIP   ONGOING

                                Bottom
F3=Exit  F4=List  F5=Refresh  F12=Cancel  F16=Wisedesk  F7=Control files
F18=Timesheet  F24=More keys
Highlighted lines indicate HELPDESK projects.
  
```

Only projects with which you are authorized to work are displayed.

Highlighted projects are Helpdesk projects, used to enter requests and problems. The rest of the projects are working projects; these are the projects you use when you are assigned a task.

- 2 Next to the project with which you want to work, type 12.

Project escalation

The project system can be set up to alert you when a request or problem is entered. Task escalation is controlled by an *escalation table*. The escalation table determines how tasks are to be processed, if they are not processed quickly enough. For example, there might be an entry for "Priority 3 (High), Status N (New)". Your policy might be that 3 N tasks must be reviewed within 2 hours and, if they have not been reviewed within 8 hours, they are escalated to priority 4.

If you use escalation tables, you can view tasks that are subject to escalation in an *escalation queue*.

For more information about escalation processing, see the *TURNOVER® Lifecycle Manager User Guide*.

MM/DD/YY 10:50:15	Work with Tasks	Your Company, Inc.
Project . . . APPR F4=List	Position to task ____	*TOP, *BOT
	Apply filters Y	F17=Select
1=Add subtask 2=Change 3=Copy 4=Delete 5=View 6=Print 9=View time		
11=Move 13=Details 14=Authority 15=Original task 20=PWM 25=Find text ...		
Task	Description	Resource Pty Status
__ 0001	Project system bug fixes	PHIPROYC 1 CLOSED
__ 0002	X-Ref Fixes	PHIPROYC 1 CLOSED
__ 0003	Fixes to copy form made in Release 3.2 (	PHIPROYC 1 DONE
20 0004	Change credit-term verify program	ELMSMIKE 1 ASSIGNED
__ 0005	Work w/ tasks-appl. column needed	PHIPROYC 1 CLOSED
__ 0006	F1 on View/Change Task not working	PHIPROYC 1 CLOSED
__ 0007	PWM opt5, then F5, parameter not passed	PHIPROYC 1 DONE
__ 0008	Need WRKPWL cmd	ELMSMIKE 1 DONE
__ 0009	Chkout panel updates	PHIPROYC 1 CLOSED
__ 0010	CHKIN panels-subfile positioning	*NONE 1 OPEN
__ 0011	Xref panel updates	ELMSMIKE 1 DONE
More...		
F3=Exit F4=List F5=Refresh F6=Add task F7=Control files F8=All projects		
F10=View 3 F11=View 2 F12=Cancel F13=Repeat F23=More options F24=More keys		

The project task key comprises:

- a 4 character project code
- a 4 character task identifier
- optionally, a 2 character subtask identifier.

The task and subtask key can be a sequentially generated number or the person entering the task can assign it.

Working with tasks

You must be authorized to Work with Projects and to the projects with which you need to work.

To	Do this
view the details of a task	Move the cursor to it, and then select option 5.
view the description of a task	Move the cursor to it, and then select option 13.
enter a new task	Move the cursor to the appropriate project entry, and then press F6. For more information, see the Help text.
change task details	Move the cursor to it, and then select option 2.
change a task description	Use F5 on the Change a Task panel.

Creating worklists

- 1 Go back to the Work with Tasks panel, and then select the task with option 20 to create a programmer worklist.

- 2 Press F3 to return to working with projects, and then select your task with option 20.

The Create Programmer Worklist panel appears.

Create Programmer Worklist (CRTPWL)

Type choices, press Enter.

List	> TP400004	Name, *SELECT
Description	*PROJECT	
Application	*PROJECT	Character value, *PROJECT
Release	—	Number
Version	—	Number
Programmer	*PROJECT	Name, *PROJECT

Bottom

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

- 3 If you are building the worklist from a task, you can use the default values, and press Enter.

Whether or not you use TURNOVER® Lifecycle Manager Projects, you must decide what to do to fulfill the request or fix the problem. You must determine which objects to change, which to recompile, what new objects you need and, of course, the specific code changes you need to make. There are many design details you must work out with your users and others in your department.

You must also gauge the impact of your changes on related objects. TURNOVER® Lifecycle Manager provides integrated cross-reference, version management, and object history panels to help you do this.

During the analysis process, you may find that some of the objects you need are checked out to someone else and you may have to coordinate your changes with another developer or wait until they have finished. If you are using the multiple checkout feature, you might be able to check out an object while someone else is working on it, but you still need to coordinate with that person to decide who will merge your changes.

Populating worklists

When you create a worklist, the Work with Programmer Worklist panel appears.

MM/DD/YY 11:25:22	Work with Programmer Worklist	Your Company, Inc.
Worklist APPR0004 Change credit-term verify program Position to . . . (Object) Apply filters . . . Y (F17=Filters) Appl/Rel/Ver . . Programmer (F4 to change) 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 Nxt Object Attr Library Obj Src Out Form FormSts ItemSts Opt		
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 No matching objects found.		

When you first create a new worklist, the subfile is blank.

The first thing to do is populate the worklist.

To	Do this
add an item (if you use TURNOVER® Lifecycle Manager's cross-reference)	Press F6. Select the item with option 15, and then review related items you may need.
go to PDM and select the items you need	Press F20, and then select option 2 to work with objects.
go to PATHFINDER or Abstract to find items you need	Press F15. For more information, see <i>'Using the multiple checkout feature' on page 70.</i>

Working with objects using PDM

Work with Objects Using PDM					YOURSYS
Library		PRODTTEST	Position to		
			Position to type		
Type options, press Enter.					
2=Change		3=Copy	4=Delete	5=Display	7=Rename
8=Display description		9=Save	10=Restore	11=Move ...	
Opt	Object	Type	Attribute	Text	
	APCLP001	*PGM	CLP	AP Tutorial Sample Program	
	APCLP002	*PGM	CLP	AP Tutorial Sample Program	
	APRPG001	*PGM	RPG	AP Tutorial Sample Program	
	APRPG002	*PGM	RPG	AP Tutorial Sample Program	
AI	APRPG991	*PGM	RPG	AP Tutorial Sample Program	
	REFCLP	*PGM	CLP	CLP Program Reference Object	
	REFRPG	*PGM	RPG	RPG Program Reference Object	
	OUTQ	*OUTQ			
					More...
Parameters or command					
===>					
F3=Exit		F4=Prompt	F5=Refresh	F6=Create	
F9=Retrieve		F10=Command entry	F23=More options	F24=More keys	
© IBM Corporation					

In PDM, use option AI (Add Item) to add items to your list. As you browse objects and source files, you can add more add items by selecting them with the AI PDM option. You can do the same thing in PATHFiNDER and Abstract. 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.

When you exit from PDM to return to your worklist, each of the items you selected appears on the list. You can refine the list by including new objects (objects you intend to create) or by removing some items after you analyze them further.

For example, you might have included all items that refer to the Vendor file. Then, after analyzing the programs, you might have removed from the list those that did not refer to the vendor payment terms field.

Once you have added items to the list, it looks like this:

```

MM/DD/YY 11:55:00      Work with Programmer Worklist      Your Company, Inc.

Worklist. . . . . APPR0004      Chg credit-term verify program
Position to . . .      (Object)      Apply filters . . . Y (F17=Filters)
Appl/Rel/Ver . .      Programmer . . . .      (F4 to change)
  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      Chk      Nxt
      Object      Attr  Library  Obj Src Out      Form      FormSts  ItemSts  Opt
      APLF001A    LF    PRODTST  Y  Y  Y      NEW      NEW      21
      APLF001A    LF    ACPTTEST  N  N  N      NEW      NEW      46
      APLF001A    LF    DEVTEST   N  N  N      NEW
      APPF001     PF    PRODTST  Y  Y  Y      NEW      NEW      21
      APPF001     PF    ACPTTEST  N  N  N      NEW      NEW      46
      APPF001     PF    DEVTEST   N  N  N      NEW
      APRPG991    PF    PRODTST  Y  Y  Y      NEW      NEW      21
      APRPG991    PF    ACPTTEST  N  N  N      NEW      NEW      46
      APRPG991    PF    DEVTEST   N  N  N      NEW
                                                    Bottom

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

```

The worklist panel tells you if the object and source actually exist in any of the libraries associated with the application with which you are working. In our example, the objects and source exist in the production (PRODTST) library, but not in development (DEVTEST) or test (ACPTTEST). Notice that none of the objects are checked out. If one had been checked out, press F11=View 2 to see the profile of the person working on it.

More about the worklist

To using the Programmer Worklist panel effectively, use the filter fields.

- 1 To display the filter fields, press F17.

```

MM/DD/YY 11:55:00      Work with Programmer Worklist      Your Company, Inc.
.....
:
:   Type filter value and Press Enter.
:
:
: Next option . . . ____      F4=List
:   Library . . . . . ____      Library name, generic*
: Level . . . . . *PROD      number, *DEV, *PROD
: Appl rel ver . . . ____ ____      F4=List
: Programmer . . . . ____      F4=List
: Item status . . . ____
: Object type . . . ____
: TurnOver Type. . . ____      F4=list
: Checked out . . . ____      Y, N, blank
:
:   F3=Exit  F4=List  F12=Cancel
:
:
:
.....
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

```

- 2 You can filter by all of these fields.

The most commonly used fields are **Next option**, **Level** (the target application level code), and **Item status**.

In the example screen above, **Level** is set to *PROD, so that you see only the objects in the production level the next time you view the worklist.

Press F1/Help for more information.

Using the worklist

A worklist is a *list of objects* upon which you intend to work or on which work is in process. It does not necessarily represent actual objects – some lines may represent objects you intend to create. Source and/or objects that exist in the associated library are highlighted. As the object progresses from level to level, it is highlighted; and the **Obj** and **Src** fields are set to Y.

By showing all levels, you can see the progress of your work from development through to production. You can tell exactly where your objects are, no matter how many levels you use for your application. Filters let you focus your attention on one level.

The PWM panel enables you to position to, filter by and perform all of your development and change management functions.

Building worklists from old forms

If a set of objects had been promoted in the past by TURNOVER® Lifecycle Manager, you can build a worklist by referring to the old form.

- 1 Look up the old form number, create a worklist, and then press F6 to add an item to the list.
- 2 Type *FORM in the **Form number** parameter and *ALL in the **Attribute** parameter.

TURNOVER® Lifecycle Manager adds all of the items to your list.

Checking cross-references

If you are using TURNOVER® Lifecycle Manager's cross-reference, you can check the cross-reference file for related items which you might need to change.

- 1 Type 15 (Check Xref) next to each production level item.

You might want to press F17 and filter by **Level** *PROD so you see only the production level.

In this example, a logical file, a physical file and a program have been selected.

The cross-reference file appears.

- The logical file is used by six programs.
- The physical file is used by the logical file and two programs.
- The program calls three programs (not all shown on the illustrated panel).

In the example, we have selected one program, APRPG993, to add to the worklist.

The **U/B** column indicates whether the related object (in the center column) uses (U) or is used by (B) the reference object (in the first column). To find out what other objects might be related to a related object, type 6 next to that item. The X-ref depth field tracks how far you have drilled down into the cross-reference database.

2 Press F17=Filters to display a panel on which to set filters for these conditions:

- *type attribute* code of related item
- items that “use” or items “used by” the reference items
- cross-reference method code
- library checked out.

Working with cross-reference objects

To	Do this
add objects to the worklist	Type 1 next to them, and then press Enter.
add objects to the worklist and check them out	Type 2 next to them, and then press Enter.
compile related objects (that is, objects you do not need to change but need to test) into your development library	Select them with option 4. The objects are created in the development library identified for that application’s first level, or if you specify *PGMR, into your own development library.

Options 1 and 2 can be submitted to batch by pressing F19. Option 4 is always submitted. You can prompt options 1, 2, and 4 by pressing F4.

If you do not check the cross-reference database

If you do not check the cross-reference database and try to create a TURNOVER® Lifecycle Manager form, TURNOVER® Lifecycle Manager displays a warning message.

If you know you do not need to check (because the change does not affect other items), you can update the worklist by typing 2 next to the items on the list and XREFCHK (Y) on the command line.

If you do not want TURNOVER® Lifecycle Manager to perform this check, press F18, page down, and then set **Require X-Ref Check** to N.

Adding new items to the worklist

When you decide to create a new object, you can add it to your worklist and then create the source and object.

To add a new item

1 Press F6.

The Add Programmer Worklist Item panel appears.

```

                                Add Programmer Worklist Item (ADDPWLITM)

Type choices, press Enter.

List . . . . . > 'APPR0004'      Name, *SELECT, *ACTIVE
Object . . . . . APLF001D        Character value, *FORM
Attribute . . . . . LF           *SELECT, *ALL, AAUDIT, ABPG...
Checkout item . . . . . *NO      *NO, *YES
Description . . . . . *OBJD

Programmer . . . . . *LIST       Name, *LIST
Reference . . . . . *LIST       *LIST, Char
User defined status . . . . . NEW Character value
Application . . . . . *LIST     Character value, *LIST
Release . . . . .              Number
Version . . . . .              Number
Create worklist . . . . . *NO    *NO, *YES
Form number . . . . . *NONE     Number, *NONE

                                Bottom
F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
F13=How to use this display   F24=More keys

```

- 2 Type of the name and attribute code of the new object you want to add to your worklist.

Usually, you can accept the defaults for the rest of the parameters. However, you may want to change the **User defined status** parameter. You can use this parameter in any way you see fit.

- 3 If you set **Checkout item** to *YES, TURNOVER® Lifecycle Manager displays an additional field, **Source copy option**. Specify how you want TURNOVER® Lifecycle Manager to handle the source you are checking out. For descriptions of the available parameters, see the online help.

When you check out the objects on your list, TURNOVER® Lifecycle Manager checks to see if the name is a valid name for your application.

Recommendation

Use **User defined status** to record your progress, especially if you are working on a long list of objects.

For example, if you are interrupted while changing a program, set the status to “more work”. When you finish programming an item, you might change the status to “done” .

The **User defined status** field can be used to mean whatever you want it to mean. When you assign a status value, you can filter by this field.

To change the field, type 2 (Change) next to an item, and then type STS(xxx), where xxx is the new status value.

Checking out objects and source

When your worklist is complete, and you are ready to start work on your objects, you need to check them out, even if you are creating new objects.

You can either:

- select and check out objects one-by-one, interactively
- submit the entire list to run to check out all of the objects at once in batch. This is the recommended method if you have more than just a few objects.

To check out all of the objects on the list

- 1 Press F17, and then do one of the following:
 - Set the **Filter by Level** to *PROD.
 - Set **Filter by next option** to 21.
- 2 Select the first object with option 21 (Checkout).
- 3 Press F13 (Repeat) to replicate option 21 throughout the entire list.
- 4 Check your session defaults to ensure that F19 is set to *BATCH.
- 5 Press F19 (Batch checkout).

MM/DD/YY 11:55:00		Work with Programmer Worklist					Your Company, Inc.			
Worklist.		APPR0004		Chg credit-term verify program						
Position to		(Object)		Apply filters . . . Y (F17=Filters)						
Appl/Rel/Ver . . .		Programmer		(F4 to change)						
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		Obj		Src		Nxt		
Object		Attr	Library	Obj	Src	Out	Form	FormSts	ItemSts	Opt
21 APLF001A		LF	PRODTEST	Y	Y	N			NEW	21
21 APLF001D		LF	PRODTEST	Y	Y	N			NEW	21
21 APPF001		PF	PRODTEST	Y	Y	N			NEW	21
21 APRPG991		RPG	PRODTEST	Y	Y	N			NEW	21
21 APRPG993		RPG	PRODTEST	Y	Y	N			NEW	21
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		

TURNOVER® Lifecycle Manager submits a job to check out all of the objects and source members. When the job is done, the source you need is in your development library.

If any problems occur during checkout, TURNOVER® Lifecycle Manager produces an error report that is put in your output queue and it sends a message to your message queue. To see the report, type WRKSPLF.

- 6 Change **Apply filters** to N, and then press F5 (Refresh).

Src and **Chk Out** change to Y, and **Nxt Opt** changes to blank.
- 7 You can edit source (option 32=Edit) as soon as it is available.
 - If you are checking out non-source objects, the **Obj** field changes from N to Y.

- If you are checking out message IDs and a message file does not exist in your development library, TURNOVER® Lifecycle Manager creates the message file and then copies the individual messages to your worklist.

Repeating similar changes

Suppose that last year you made a change similar to one you are about to make. View the TURNOVER® Lifecycle Manager form used last year. From the view panel, you can add all of the objects to your worklist and check out all of the objects at the same time. To view an old form, do one of the following:

- Find last year's task in the project file, and then view the related form.
- Use the Work with Object History panel (option 18 on the worklist or command `TURNOVER 4`) to find one object changed last year, and then display the form.
- Find the form on Work with TURNOVER® Lifecycle Manager forms (command `TURNOVER 6`), and then display it.
- Press F6 on the Worklist panel, enter `*FORM` in the **Form number** field, type the form number, and then type `*ALL` in the **Attribute** field.

Change history information is available only if it has not been purged from TURNOVER® Lifecycle Manager's files. Keep change history online for as long as possible.

What happens during checkout

For example, assume that you are checking out source for an object.

- 1 You identify an object along with its type (or attribute) and application code. (These all were done when the item was added to your worklist.)
- 2 TURNOVER® Lifecycle Manager looks at the application definition for the library name, and then finds the object.
- 3 TURNOVER® Lifecycle Manager finds the source from the application definition.
- 4 If the source is not where the application says it is, TURNOVER® Lifecycle Manager looks at the object description.
- 5 TURNOVER® Lifecycle Manager checks that the source and object match. If they do not, it issues a warning.

If the global parameter **Level checking on checkout** is `Y`, you cannot check out the object until the discrepancy is resolved. On the main menu, select option 8, and then option 5.

- 6 TURNOVER® Lifecycle Manager copies the source to your development library.

TURNOVER® Lifecycle Manager decides which library to use for checkout by looking at the highest unlocked level for the application definition; level 2 in our AP application.

It decides where to copy the source by referring to the **Source From library** field of the lowest level of the Application definition.

- If it is set to *PGMR, TURNOVER® Lifecycle Manager looks at your programmer enrollment record (Main Menu option 2).
- If it is set to *PROJECT, TURNOVER® Lifecycle Manager looks at your project record to determine the library name.

If the source is checked out already, TURNOVER® Lifecycle Manager does not allow you to check it out again (except as an emergency change: see *'Emergency changes' on page 68*).

If you are checking out source interactively and a source member exists in the development library with the same name, you are asked whether or not to overlay it. If you are checking out in batch, TURNOVER® Lifecycle Manager will not overlay a source member; a message is printed on the Checkout Report and the object is recorded as having been checked out.

If the object is checked out from production and exists in an intermediate level (ACPTTEST in our AP example), TURNOVER® Lifecycle Manager prompts you to check it out of that library. If you are checking out in batch, TURNOVER® Lifecycle Manager checks out the object from the intermediate level.

TURNOVER® Lifecycle Manager verifies that the source and object match by examining its internal database (for information about when the object was created and when its source last changed). It examines the object description and member description. If the object does not match the internal record, you will get a warning message. This warning indicates that there is a possible discrepancy you should verify to be sure you have the correct source.

- If you are checking out a non-source object, TURNOVER® Lifecycle Manager writes a record in the checkout file and duplicates the object to your development library. If you are changing it, you can recreate it in test to reflect your change. When you promote it, it is moved or duplicated back into production.
- If you are checking out a source-only object, such as /COPY code or S/36 OCL, TURNOVER® Lifecycle Manager recognizes that condition and copies the source to your development library.
- If you are checking out an object that does not exist, TURNOVER® Lifecycle Manager reserves that name for you. See *'Reserving new object names' on page 39*.
- If you check out in batch, review the checkout report for any discrepancies. TURNOVER® Lifecycle Manager sends you a message if it finds significant discrepancies.

Note When the administrator defines your applications, they can run a job that verifies, among other things, that source and objects match. Several reports are produced. Discrepancies should be resolved at that time. If you see a discrepancy, see your TURNOVER® Lifecycle Manager Administrator. Make sure you have the correct source before you begin to work.

More information

For information about checking out or reserving other items, see *'Message IDs, message files, menus, and other object types' on page 82*.

Reserving new object names

When you add a new object name to your worklist (see *'Adding new items to the worklist' on page 34*), you must identify (reserve) that name to TURNOVER® Lifecycle Manager. To reserve it, select the object name with option 21 (Checkout). If the name does not exist in your production library or it has not already been reserved, TURNOVER® Lifecycle Manager reserves the name to you. The name must conform to naming standards set up for your application.

Reserving an object name is similar to checking it out, except that no source is copied from the production library.

Coding and testing changes

You can continue working on the PWM panel to develop the objects on your list. If you are familiar with PDM, you will find that many of the functions are similar. However, with PWM you see just the objects on which you need to work to complete your task. You can perform all of the operations you need to do on the Worklist panel.

- 1 Set **Apply filters** to N (as illustrated below), or press F17 (Filters), and then set the **Filter by Level** to *DEV to filter only the development level objects.

MM/DD/YY 11:55:00		Work with Programmer Worklist					Your Company, Inc.			
Worklist.		APPR0004		Chg credit-term verify program						
Position to . . .		(Object)		Apply filters . . .		N		(F17=Filters)		
Appl/Rel/Ver . .		Programmer				(F4 to change)				
15=Check X-ref		16=Related objects		18=Object history						
32=Edit source		35=Browse source		36=Compile		37=SDA/RLU				
		Object		Chk		Nxt				
	Object	Attr	Library	Obj	Src	Out	Form	FormSts	ItemSts	Opt
	APLF001A	LF	PRODTEST	Y	Y	Y			NEW	
__	APLF001A	LF	ACPTTEST	N	N	N			NEW	46
32	APLF001A	LF	DEVTEST	Y	N	N			NEW	32
	APPF001	PF	PRODTEST	Y	Y	Y			NEW	
__	APPF001	PF	ACPTTEST	N	N	N			NEW	46
32	APPF001	PF	DEVTEST	Y	N	N			NEW	32
	APRPG991	RPG	PRODTEST	Y	Y	Y			NEW	
__	APRPG991	RPG	ACPTTEST	N	N	N			NEW	46
32	APRPG991	RPG	DEVTEST	Y	N	N			NEW	32
	APRPG993	RPG	PRODTEST	Y	Y	Y			NEW	
										More...
==>										
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		

- 2 You can do all of your development work on the Worklist panel.

Nxt Opt (Next Option) for the items in the development library is set to 32.

To edit the source of an object, type 32 (Edit source) next to it.

For display or printer files, type 37 (SDA/RLU).

- 3 You can filter the level code by *DEV and *PROD. During this phase, the recommended filter is by *DEV.

Worklist options

Key	Description
2 Change	Change information in the worklist entry.
3 Copy	Copy this item to another worklist.
4 Delete	Delete an item from worklist.
11 Move	Move an item from one worklist to another.
12 Compare source	Compare the source of the item you select with any other version of source available.
15 Check X-Ref	Check for related entries in the cross-reference database.
16 Related objects	Check for related objects in related applications.
18 Object history	View history for an object. (You will see the Work with Object History panel.)
21 Checkout	Check out the object from production or an intermediate level.
22 Change checkout	Change the checkout record. For example, you can reassign the checkout to another person or change the attribute code.
24 Check-in	If it is checked out in error or your requirements change and you no longer need to work on the item on your worklist.
25 Task details	Change information in or view task details for the current task.
26 Conflicts	To view and resolve object version conflicts. See 'Managing Multiple Versions' in the <i>TURNOVER® Lifecycle Manager Administrator's Guide</i> .
32 Edit source	Edit a source member using SEU.
35 Browse source	Browse a source member in SEU.
36 Compile	Compile using command defined in TURNOVER® Lifecycle Manager for the object type selected.
37 SDA/RLU	Run SDA or RLU. TURNOVER® Lifecycle Manager runs whichever is appropriate.
38 Pre/post-Cmds	To enter commands to be linked to the object and run before or after the object is created. Commands that are linked are carried forward as the object is promoted.
40 Form jobs	See all of the jobs related to promoting objects.

Key		Description
41	Error check	Error check form to see if it is complete.
42	Edit form	View and edit an existing form.
43	Copy form	Create a new form from an existing form.
44	Remove from form	Take an item off a form if it was added in error or is no longer needed.
45	View form	Look at the form for a worklist entry.
46	Add to form	Add this item to a TURNOVER® Lifecycle Manager form.
47	Run form	Submit a form for processing (READY status only).
48	Approve form	Approve a form, or review the approval list. Form status is APL - PN.
49	Print form	Print a hard copy of a TURNOVER® Lifecycle Manager form.

You can define your own user options by pressing F16=User options.

Worklist function keys

Key		Description
F2	Forms	View all forms associated with your worklist.
F3	Exit	Exit from worklist.
F4	Prompt	List valid values when cursor is positioned on a field, or to prompt a selection or command.
F5	Refresh	Redisplay panel to view changes.
F6	Add item	Add a item to the list.
F7	Previous worklist	Switch quickly to the last worklist you worked on.
F9		Retrieve last command keyed on this display.
F10	Command entry	Switch to the IBM i command entry panel.
F11	View 2	View additional information in the subfile.
F12	Cancel	Equivalent to exit.
F13	Repeat	Copy an option code to the remaining items on the list when an option is entered next to an item.

Key		Description
F14	WRKSBMJOB	To execute the work submitted jobs command.
F15	Customized Fkey	Your Administrator can customize this fkey for your company or for individual users (using the CUSTFKEY command). This is frequently set to provide easy access to PATHFiNDER or Abstract, if you have one of these products.
F16	User options	Enter or change your user options.
F17	Filters	Work with the window to set Worklist Panel filters values.
F18	Defaults	To view and change session defaults.
F19	Inter. 21/46	Select object on list with option 21, then press Enter to submit the checkout to batch or press F19 to run it interactively. Works similarly for option 46. (Press F18 for a session default to control how the F19 key works.)
F20	PDM	Run the start PDM (STRPDM) command.
F21	Print list	Print a list of the items on the worklist.
F22	Add time	Add time to your time sheet for the project task associated with your worklist.

Compiling the objects

After you have edited the source, you can select and compile the objects into the development library by typing `36 (Compile)` next to each item in your development library.

```

MM/DD/YY  12:02:12      Work with Programmer Worklist      Your Company, Inc.

Worklist. . . . . APPR0004      Chg credit-term verify program
Position to . . .      (Object)      Apply filters . . . N (F17=Filters)
Appl/Rel/Ver . .      Programmer . . . .      (F4 to change)
15=Check X-ref 16=Related objects 18=Object history
32=Edit source 35=Browse source 36=Compile 37=SDA/RLU 38=Pre/post-Cmds

      Object                      Chk                      Nxt
Object      Attr      Library      Obj      Src      Out      Form      FormSts      ItemSts      Opt
  APLF001A      LF      PRODTST      Y      Y      Y                      NEW
  APLF001A      LF      ACPTTEST      N      N      N                      NEW      46
36 APLF001A      LF      DEVTEST      Y      N      N                      NEW      32
  APPF001      PF      PRODTST      Y      Y      Y                      NEW
  APPF001      PF      ACPTTEST      N      N      N                      NEW      46
36 APPF001      PF      DEVTEST      Y      N      N                      NEW      32
  APRPG991      RPG      PRODTST      Y      Y      Y                      NEW
  APRPG991      RPG      ACPTTEST      N      N      N                      NEW      46
36 APRPG991      RPG      DEVTEST      Y      N      N                      NEW      32
  APRPG993      RPG      PRODTST      Y      Y      Y                      NEW

More...

Parameters or command
====>
F7=Previous Worklists F12=Cancel F13=Repeat F14=WRKSBJOB F15=PATHFINDER
F16=User options F17=Filters F18=Defaults F20=PDM F24=More keys

```

Using create command overrides

If you override the create command for an object by prompting option 36 (Compile) and changing a command parameter, TURNOVER® Lifecycle Manager remembers the override and asks if you want to use it whenever you subsequently compile the object.

The Overridden Creation Command Confirmation panel appears.

Overridden Creation Command Confirmation

Object ACTPYBMSTF
Library MARYSDEV
Type *FILE

The creation command was overridden the last time that the above object was created. Select the command that should be used to create the object, and press Enter. If F4 was pressed from the previous panel, the selected command will be prompted. To exit without creating the object, press F12.

1=Select
Overridden Command
CRTPF FILE(MARYSDEV/ACTPYBMSTF) SRCFILE(MARYSDEV/QDDSSRC) SRCMBR(ACTPYBMSTF)
GENLVL(19) EXPDATE(090320)

Default Command
CRTPF FILE(MARYSDEV/ACTPYBMSTF) SRCFILE(MARYSDEV/QDDSSRC) SRCMBR(ACTPYBMSTF)
GENLVL(19)

F12=Cancel

Customizing the worklist session defaults

The session default control how objects are created. You control how TURNOVER® Lifecycle Manager creates objects in your development library, and many other session features. These defaults are saved by user profile.

To display the session defaults, press F18.

Change Session Defaults: 1

MM/DD/YY 12:02:12	Work with Programmer Worklist	Your Company, Inc.
:	Change Session Defaults	:
:	:	:
:	Save session defaults Y	Y, N
:	Default option 21/46 mode *BATCH	*BATCH, *INTER
:	Edit/compile authority *DEVELOP	*SYSAUTH,*DEVELOP,*CHECKOUT
:	:	:
:	Test object creation parameters	:
:	Link option 36 compiles Y	Y, N
:	Job description *USRPRF	*USRPRF, name
:	Library	Library name
:	Job queue *JOB	*JOB, name
:	Library	Library name
:	Output queue *JOB	*PGMR, *JOB, name
:	Library	Library name
:	Library list *APP	*APP, *CURRENT
:	Replace object Y	Y, N, A
:	:	More...
:	F3=Exit F12=Cancel	:
:	:	:
:	:	:
:	:	:

Save session defaults

To save your current session defaults when you exit from the programmer worklist, specify **Y**.

Default option 21/46 mode

This parameter controls how the F19 key works on the worklist.

Value	Description
*BATCH	Pressing Enter runs options 21 or 46 in batch.
*INTER	Pressing Enter runs options 21 or 46 interactively.

When you press the **F19** key, the opposite action occurs.

The recommended value is ***BATCH**.

Edit/compile authority

Option	Description
*SYSAUTH	Your access to objects and your ability to delete, change, or create objects is controlled by your IBM i system authority.
*DEVELOP	You can create or work with objects only in your development library. This prevents you from deleting, creating, or changing objects in libraries other than development, even if you have authority to those libraries and objects.

Option	Description
*CHECKOUT	You can edit and compile an object only if you have the lowest checkout record for that object.
	You can only edit and compile the object in your development library.

This parameter relates to options 32 (Edit source), 36 (Compile), and 37 (SDA/RLU). This parameter does not relate to what you can do on the command line.

The recommended value is *DEVELOP.

Test object creation parameters

Use **Job description**, **Job queue**, and **Output queue** to control how your items are created. These parameters apply to option 36 (Compile), and are used as overrides when submitting compiles.

Link option 36 compiles

This parameter tells TURNOVER® Lifecycle Manager whether a series of objects you select with option 36=Compile should be linked. Linking the compiles means that compiling one of any of the objects selected depends on the successful completion of the processing attempt for the previous object in the series.

Value	Description
Y	Link the compiles
N	Do not link the compiles

The default value is Y.

Objects can have pre- or post-run commands associated with them. Under these circumstances, TURNOVER® Lifecycle Manager conditions its attempt to continue processing on the Error recovery parameter specified for the command, as shown here:

When compiles are linked

Error condition	Recovery	Result
Preform cmd fails	G=Go	Go on to remaining pre cmds
Preform cmd fails	C=Cancel	End the job
PreLine cmd fails	G=Go	Go on to remaining pre cmds
PreLine cmd fails	C=Cancel	End the job
Compile fails	n/a	End the job

Error condition	Recovery	Result
PostLine cmd fails	G=Go	Go on to remaining post cmds
PostLine cmd fails	C=Cancel	End the job
PostForm cmd fails	G=Go	Go on to remaining post cmds
PostForm cmd fails	C=Cancel	End the job

When compiles are not linked

Error condition	Recovery	Result
Preform cmd fails	G=Go	Go on to remaining pre cmds
Preform cmd fails	C=Cancel	End the job
PreLine cmd fails	G=Go	Go on to remaining pre cmds
PreLine cmd fails	C=Cancel	Stop processing this object; begin processing next object
Compile fails	n/a	Stop processing this object; begin processing next object
PostLine cmd fails	G=Go	Go on to remaining post cmds
PostLine cmd fails	C=Cancel	Stop processing this object; begin processing next object
PostForm cmd fails	G=Go	Go on to remaining post cmds
PostForm cmd fails	C=Cancel	End the job

If an object's preline or postline command fails and the Error recovery parameter is c (Cancel), or if the compile fails for an object and the object is not one of a linked series, the postform commands for the object are not attempted.

Library list

Option	Description
*CURRENT	TURNOVER® Lifecycle Manager uses your current library list when you select option 36=Compile to submit a job to compile objects. You are responsible for setting up your interactive library list appropriately.

Option	Description
*APP	TURNOVER® Lifecycle Manager constructs a library list consisting of your development libraries, followed by the library list defined at the first (lowest) level of your application definition.

The recommended value is *APP.

Replace object

If you select option 36 (Compile) and the object already exists in your development library, TURNOVER® Lifecycle Manager displays a confirmation panel with the **Replace** parameter set to Y or N, depending on the value you set for this parameter.

Set this parameter to A to bypass the confirmation panel and automatically replace existing objects.

The recommended values are Y or A.

Change Session Defaults: 2

MM/DD/YY 12:02:12	Work with Programmer Worklist	Your Company, Inc.
.....		
:	Change Session Defaults	:
:		:
:	Save session defaults Y	Y, N
:		:
:	List sequence *OBJECT	*OBJECT, *LIBRARY, *TYPE...
:	User option file TUSROPTF	File name
:	Library *LIBL	Library name
:	Position-to default *TOP	*TOP, *ITEM
:	Show perm-lock levels N	Y, N
:		:
:	Require X-Ref check N	Y, N
:	X-Ref automatic Y	Y, N
:	Option 15 library to X-Ref *ALL	*SELECTED, *ALL
:	Require Related apps check Y	Y, N
:	Related apps check automatic . . . Y	Y, N
:		:
:		More...
:	F3=Exit F12=Cancel	:
:		:
:		:
:.....		

List sequence

Specifies the order of items in the worklist.

Option	Description
*CRTSEQ	By logical create sequence: physical files precede logical files that precede programs, and so on
*OBJECT	By object name

Option	Description
*LIBRARY	By object name within library name
*TYPE	By type attribute code, object name and library.

The recommended value is *OBJECT.

User option file

You can create and maintain your own set of user options, available when you press F16 on your worklist.

To create your own option file, copy the TUSROPTF file from the SOFTTURND library into your own library, and then change **User option file** to the name of your copy and its library.

Position-to default

Value	Description
*TOP	Reposition the worklist subfile to the top when you change filters.
*ITEM	The worklist subfile to maintains its position when you change filters.

Show perm-lock levels

If you are maintaining your own modifications of a vendor-supplied application, you might have defined additional application definition levels to define your vendor's unchanged objects and source.

To see these on your Worklist panel, specify Y; otherwise, specify N.

Require X-Ref check

TURNOVER® Lifecycle Manager prevents you from creating or running a form until you have performed a cross-reference check (see ['Checking cross-references' on page 32](#)).

To display a reminder to check the cross-reference file, specify Y.

If you know you do not have to check the cross-reference, you can change the XREFCHK worklist item field to Y.

The recommended value is Y.

X-Ref automatic

If you want TURNOVER® Lifecycle Manager to search the cross-reference and add any items not already on the worklist to the TURNOVER® Lifecycle Manager form to be recompiled in the target library, specify Y.

If you want to view and add items yourself when you build the TURNOVER® Lifecycle Manager form, specify N.

The recommended value is Y.

Option 15 library to X-Ref

This parameter tells TURNOVER® Lifecycle Manager what library to look in when checking the cross reference for objects that use, or are used by, an object you select with option 15.

Option	Description
*SELECTED	TURNOVER® Lifecycle Manager shows only references from the object library associated with the object you selected.
*ALL	TURNOVER® Lifecycle Manager shows references from all libraries in the cross-reference file.

Require Related apps check

TURNOVER® Lifecycle Manager prevents you from creating or running a form until you have checked for objects in related applications.

To display a reminder to check for related application objects, specify Y.

If you know you do not have to check for related objects, you can change the RELAPPCHK worklist item field to Y.

For more information, see “Working with application definitions” in the *TURNOVER® Lifecycle Manager User Guide*.

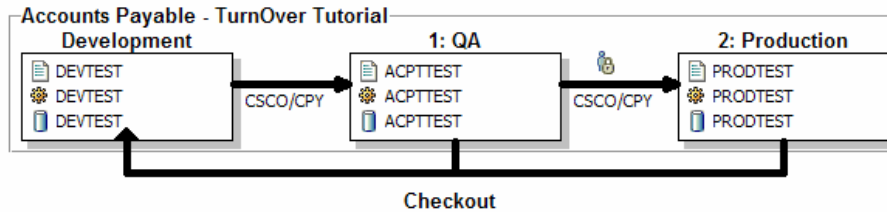
Related apps check automatic

If you want TURNOVER® Lifecycle Manager to search for related application objects and create TURNOVER® Lifecycle Manager forms for related applications, and add these items to your related forms if they are not already on those forms, specify Y.

If you want to view and add items yourself, when you build the TURNOVER® Lifecycle Manager form, specify N.

Creating TURNOVER® Lifecycle Manager forms

After you have completed programming and testing your changes, you can create a TURNOVER® Lifecycle Manager form to promote your objects. In TURNOVER® Lifecycle Manager, *form* means *promotion request*.



A form is all the information TURNOVER® Lifecycle Manager needs to promote objects and source from left to right. Checkout occurs *right to left* and is done when you start a programming task. TURNOVER® Lifecycle Manager promotions occur when you are done programming and are ready to have your work certified and put back into production.

Contents of a form

A TURNOVER® Lifecycle Manager form consists of:

- *job control information* (the form header)
- *object information* (the lines on the form).

A form can also contain:

- **Commands** (pre-run or post-run). Commands like override database (OVRDBF) file necessary for creating some objects can be included. Commands can be associated with particular objects on the form. When you do this, TURNOVER® Lifecycle Manager remembers the association the next time you or someone else changes the object.
- **Create command overrides**. You can override the default create command TURNOVER® Lifecycle Manager uses to create this object. This override is associated with the object on the form. When you do this, TURNOVER® Lifecycle Manager remembers the command override the next time you or someone else changes that object.
- **Comments** (form and/or line). These provide additional documentation about the form. Sometimes comments are used for instructions to the Administrator or to operations.
- **Lock checks** (objects that are checked for locks before the form starts). When you include objects to be checked, TURNOVER® Lifecycle Manager checks locks and allocates the object, if you want it to. You can check locks on any objects, not necessarily those you are changing now. You can use the **Automatic form lock checking** global default to automatically check for locks on files only or on all data objects before TURNOVER® Lifecycle Manager replaces such objects during a form run. For more information, see "Setting global defaults" in the *TURNOVER® Lifecycle Manager User Guide*.
- **Completion messages**: Messages can be set up to be sent if the form runs successfully, fails or in either case. Default messages can be set up for your application definition. Completion messages are always sent to the developer and designated administrator, without having to be specified explicitly on the TURNOVER® Lifecycle Manager form.

More information about forms

- A form is identified by a unique seven-digit number. The number comprises a two-digit form prefix and a five-digit sequence number assigned by TURNOVER® Lifecycle Manager.

The maximum number of forms in a system is 99,999.

- Each form contains a reference to your project task (subject to how your environment is set up). This field is called **Reference** or **Project task reference**.
- Each form is run once. If a form needs to be rerun, it is copied to create a new form, and the new form is run. This enables TURNOVER® Lifecycle Manager to maintain a complete audit log.
- The objects on a form are sequenced such that objects are created in the correct logical order: PF first, then LF, then DSPF, and so on.
- To reserve space for 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 is 900.
- Logical files are added to the form automatically for physical files in a target library. You need to identify explicitly only those logical files that you have changed.
- Forms can be deleted by the TURNOVER® Lifecycle Manager administrator. Be careful not to remove valuable change history.
- Form status determines whether a form can be run. Only forms that are `READY` or `RECVD` status can be run. (`RECVD` is valid only on remote production systems where TURNOVER® Lifecycle Manager changes have been “received.”)
- If form status is `APL-PN` (Approval Pending), the form must be approved before it can be run.
- When distributed to another system, a form retains its original form number.
- Forms can be scheduled to run at any time in the future.
- Forms can be grouped together and run as a group or run independently. They can also be linked such that one form must complete successfully before another form can start.
- If you use field reference files, you can identify them on a TURNOVER® Lifecycle Manager form as type `PFREF`. TURNOVER® Lifecycle Manager then makes sure that they are created first. When you check out the file, prompt the checkout option and change the **Override attribute** field to `PFREF`; or type the `DSPCHKOUT` command, and then change the **New type code** field to `PFREF`.

Creating forms

When you are ready to promote source and objects to production or to some intermediate test level, you must create a TURNOVER® Lifecycle Manager form.

- 1 Press F17 to set the **Filter by level** to 1 (the acceptance test level in our AP application) or the **Filter by next option** parameter to 46.

The Work with Programmer Worklist panel appears.

MM/DD/YY 12:05:15		Work with Programmer Worklist		Your Company, Inc.	
Worklist APPR0004		Chg credit-term verify program			
Position to		(Object) Apply filters . . . Y (F17=Filters)			
Appl/Rel/Ver		Programmer (F4 to change)			
40=Form jobs		41=Error check		42=Edit form	
43=Copy form		44=Remove from form		45=View form	
46=Add to form		47=Run form		48=Form approval	
49=Print form					
Object		Attr		Library	
Obj		Src		Out	
Form		FormSts		ItemSts	
Opt					
46	APLF001A	LF	ACPTTEST	N	N
46	APPF001	PF	ACPTTEST	N	N
46	APRPG991	RPG	ACPTTEST	N	N
46	APRPG993	RPG	ACPTTEST	N	N
Bottom					
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					

- 2 Type 46 next to the first object.

The **Nxt Opt** field should be 46=Add to Form.

- 3 Press F13 (Repeat) to repeat option 46 throughout your list.
- 4 Press Enter to submit the form build to batch.

TURNOVER® Lifecycle Manager builds a form and adds each of the items you selected to that form.

Note The default action of the Enter key and F19 can be reversed. Press F18 (Defaults) for session defaults.

- 5 Press F5. The **Form number**, **Form Sts**, and **Nxt Opt** columns change.
- 6 If you want to edit the form after it has been created, select the first object on the list with option 42 to view and check the form.

Viewing and editing a form

Select any object with option 42 (Edit form) or 45 (View form).

```

MM/DD/YY  12:05:15      Work with Programmer Worklist      Your Company, Inc.

Worklist. . . . . APPR0004      Chg credit-term verify program
Position to . . . . . (Object)      Apply filters . . . Y (F17=Filters)
Appl/Rel/Ver . . . . . Programmer . . . . . (F4 to change)
40=Form jobs  41=Error check  42=Edit form  43=Copy form  44=Remove from form
45=View form  46=Add to form  47=Run form  48=Form approval  49=Print form

      Object      Chk      Nxt
      Object      Attr  Library  Obj Src Out  Form  FormSts  ItemSts  Opt
42  APLF001A  LF    ACPTTEST  N  N  N    0300023  Ready  NEW    47
      APPF001  PF    ACPTTEST  N  N  N    0300023  Ready  NEW    47
      APRPG991  RPG    ACPTTEST  N  N  N    0300023  Ready  NEW    47
      APRPG993  RPG    ACPTTEST  N  N  N    0300023  Ready  NEW    47

Bottom

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

```

You can edit the form to include pre- or post-run commands, comments, lock checks, or special completion messages. (As the programmer, you will always receive a message after the form runs.)

```

MM/DD/YY      Form Maintenance      Your Company, Inc.
12:30:23

Form . . . . . 300027      Chg credit-term verify program
Application . . . . . AP      Rel/Ver/Lev . . . . . 1
Job description . . DEVELOP      Output queue . . . . . ELMSMIKEOQ
      Library . . . . . ACPTTEST      Library . . . . . ELMSMIKE
Library list . . . . . DEVELOP08      Job accounting . . . . .
Programmer . . . . . ELMSMIKE      Reference . . . . . APPR0004
Form comments: N Lock checks: N      Completion messages: N Archive objects: N
Position to line . . . . . or Object . . . . . Documents: N
2=Change 4=Delete 7=Comments 8=Pgming time      M=Move B=Before A=After
10=CreateParms 11=PreRunCmd 12=PostRunCmd      Source Libraries

      Ln Object      Library      Type  Cd  FROM Lib  FROM      TARGET      Cmts
___ 001 APPF001      ACPTTEST  PF    R  ELMSMIKE  ELMSMIKE  ACPTTEST  N
___ 002 APLF001A      ACPTTEST  LF    R  ELMSMIKE  ELMSMIKE  ACPTTEST  N
___ 003 APLF001D      ACPTTEST  LF    R  ELMSMIKE  ELMSMIKE  ACPTTEST  N
___ 004 APLF991      ACPTTEST  RPG    R  ELMSMIKE  ELMSMIKE  ACPTTEST  N
___ 005 APLF993      ACPTTEST  RPG    R  ELMSMIKE  ELMSMIKE  ACPTTEST  N

F3=Exit  F4=Comments  F5=Resequence  F6=Add line  F7=Locks  F24=More keys
F22=File select  F10=Documents  F12=Cancel  F13=Other pgmr chgs

```

Options (in edit mode)

Option	Description
2 Change	View or edit the form line.
4 Delete	Delete the form line.
7 Comment	Enter or edit comments about the form line.
10 CreateParms	Override the default creation parameters for the item on this line.

Option	Description
11 Pre-run commands	Commands associated with an object, to be run during form processing before objects are created. For example, <code>OVRRDBF</code> , where program uses temporary file.
12 Post-run commands	Commands associated with an object, to be run after an object is created or during the finalization phase. Example: call to your file conversion program; where the default <code>CPYF</code> command is inappropriate.

Functions

Function	Description
F4 Comment	Enter form comments.
F5 Resequence	Resequence the objects on the form. Objects are added to the form in the correct logical order (PF, LF, DSPF, RPG, and so forth) for running the form.
F6 Add line	Add another object to the form. (We recommend that you add the object to your worklist first, then add that object to the form using option 46.)
F7 Lock checks	Supply names of objects you want TURNOVER® Lifecycle Manager to check for locks or allocate before changing objects on the form. Specify <code>*ALL</code> to check for locks on all files (specify <code>*FILE</code> for the object type parameter) or all data objects (specify <code>*DTAOBJ</code> for the object type parameter) before it replaces such objects during the form run. TURNOVER® Lifecycle Manager then checks the objects for locks before starting the form run and sends a message to QSYSOPR if it finds any locks on the specified objects. The operator can clear the locks and continue the form run or cancel the form. If the operator cancels the form, it returns to a <code>READY</code> status.
F8 Comp. Message	Enter completion messages. Programmer and Administrator always receive a message.
F9 Insert line	Adds a line immediately following the cursor position.
F11 Header	To view or edit the form header information (the information at the top of the panel).
F12 Cancel	Exits the form.
F16 User options	View or change user options defined for this panel.

Function	Description
F18 Insert command	Insert a pre-run or post-run command. Position just above first line to insert pre-run commands; at last line to insert post-run commands.

Press F1/Help for more information.

Form Maintenance Options panel

When you exit from a TURNOVER® Lifecycle Manager form, the Form Maintenance Options panel appears.

```
MM/DD/YY 12:35:55      Form Maintenance Options      Your Company, Inc.

To run Form Maintenance exit options, type choices and press Enter. To return
to Form Maintenance without running options, press F12. To exit without
running options, press F3.

Form . . . . . 5300225
Description . . . . . Test form

Resequence the form . . . . . N Y, N

Check for logical files and add them to the form . . . N Y, N

Check cross-reference file for related objects . . . . N Y, N

Check for objects changed by other programmers . . . . N Y, N
Check for related application conflicts . . . . . Y Y, N

F3=Exit  F12=Cancel
```

- 1 Normally, you accept the defaults set by TURNOVER® Lifecycle Manager and press Enter. If you have already performed these operations, you can avoid doing so again by typing N in these fields.

If you do not want to run any exit options, do one of the following:

- Press F12 to return to the TURNOVER® Lifecycle Manager form Maintenance panel.
- Press F3 to return to the Work with TURNOVER® Lifecycle Manager forms panel.

- 2 After you specify your exit options, press Enter to run them. When TURNOVER® Lifecycle Manager finishes running the options, it returns to the Work with TURNOVER® Lifecycle Manager forms panel.

Resequence the form

This puts the form in the correct logical order (PFREF, PF, LF, DSPF, Programs, and so on).

To view the file that contains the TURNOVER® Lifecycle Manager type codes and the sequence field that controls this operation, type `TURNOVER 8` on a command line, then option 4.

Normally, the form is created in the correct logical order.

Check for logical files and add them to the form

TURNOVER® Lifecycle Manager checks the form for any logical files over physical files being replaced or deleted, and adds them to your form automatically.

Check cross-reference file for related objects

If you are using TURNOVER® Lifecycle Manager's cross-reference function or Hawkeye's PATHFINDER or HelpSystems' Abstract, you can check to see if there are related objects you need to include on the form for recompilation. You can add them to your form. This may have been done automatically; see *'Using create command overrides' on page 43*.

Check for objects changed by other programmers

If you are using the Multiple Checkout feature, another developer might have changed an object on your list while you were working on it. See *'Using the multiple checkout feature' on page 70*.

Check for related application conflicts

If you are using the TURNOVER® Lifecycle Manager related application feature, you must make sure there are no conflicts before you promote the objects. For more information, see "Setting up additional TURNOVER® Lifecycle Manager environments" in the *TURNOVER® Lifecycle Manager User Guide*.

Checking forms for errors

If you do not have authority to run a form or you do not want to submit the form right away, you can pre-edit the form to determine if it is set up correctly.

Select any object with option 41 (Check form). This operation can be done interactively or in batch. If you do it in batch, a brief report of error-conditions encountered is produced. If you do this interactively, then you can view messages of any error-conditions.

Note If you are going to submit the form to be run, a pre-run error-check is done at that time; it is not necessary to do it beforehand.

Alternate path: On a command line, type `TURNOVER 6` or select option 6 on the TURNOVER® Lifecycle Manager Main Menu to view all TURNOVER® Lifecycle Manager forms. You can filter and select your forms with option 1 to error-check your form.

For more information about what TURNOVER® Lifecycle Manager checks during the pre-run error-check, see "Submitting forms to run" in the *TURNOVER® Lifecycle Manager User Guide*.

Getting forms approved

If the form status is **APL-PN** (Approval Pending), you can send messages to the people on the approval list to prompt them to approve the form. Select any item with **Nxt Opt** of 48 and follow the instructions on the panel. Forms with **APL-PN** status must be approved before they are run.

```
MM/DD/YY 12:05:15      Work with Programmer Worklist      Your Company, Inc.

Worklist . . . . APPR0004      Chg credit-term verify program
Position to . . .      (Object)      Apply filters . . . Y (F17=Filters)
Appl/Rel/Ver . .      Programmer . . . .      (F4 to change)
40=Form jobs  41=Error check  42=Edit form  43=Copy form  44=Remove from form
45=View form  46=Add to form  47=Run form  48=Form approval  49=Print form

      Object      Chk
      Object      Attr  Library  Obj Src Out  Form  FormSts  ItemSts  Nxt
48  APLF001A  LF  ACPTTEST  N  N  N  0300027  APL-PN  NEW  48
    APPF001  PF  ACPTTEST  N  N  N  0300027  APL-PN  NEW  48
    APRPG991  RPG  ACPTTEST  N  N  N  0300027  APL-PN  NEW  48
    APRPG993  RPG  ACPTTEST  N  N  N  0300027  APL-PN  NEW  48

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

Bottom
```

Messages are sent to each of the people on the approval list. They must select **TURNOVER®** Lifecycle Manager main menu option 6, press F10 to view the forms that are awaiting their approval, and then select the form with option 10. They can view the form and view the project task description before approving the form.

For more information about approving forms, see “Working with **TURNOVER®** Lifecycle Manager forms” in the *TURNOVER® Lifecycle Manager User Guide*.

Running forms

If your **TURNOVER®** Lifecycle Manager administrator (or someone else) is responsible for running forms, then your job is finished when you create the form and notify the people on the approval list or the administrator that the form is ready.

If the form is not approved or if, after it runs, programming errors are discovered during testing, then you are responsible for resolving the errors.

To run a form

If you are responsible for running your own **TURNOVER®** Lifecycle Manager forms, then you can submit the form to run by doing the following:

- 1 Select the first item on your worklist with option 47.

If you select more than one TURNOVER® Lifecycle Manager form to run, you'll see another panel asking you if you want to run the forms as a group. For more about group forms, see the *TURNOVER® Lifecycle Manager User Guide*.

The Submit Form panel appears.

MM/DD/YY 14:20:30	Submit Form	Your Company, Inc. SYSTEM: YOURSYS
Form 300027	Chg credit-term verify program	
Application AP	Release:	Version: Level: 1
Status READY		
Type options, press Enter.		
1=Select		
Opt	Form Submit Option	
1	Error-check in batch	
-	Error-check interactively	
-	Submit to run now	
1	Submit to run as timed job	Date: 5/15/20 Time: 0400 hhmm
-	Submit on hold	
-	Work with form	
-	Work with form distribution defaults	
F3=Exit F5=Refresh F10=Form approval F12=Cancel		

This panel is the same as the one you would see if you had selected option 7 on the Work with forms panel.

- 2 TURNOVER® Lifecycle Manager always does a pre-run error-check before it runs your form. On the Form Submit panel, you can choose to:

- Do the pre-run error-check interactively or in batch
- and
- Submit the TURNOVER® Lifecycle Manager job to run immediately, or
- Submit the TURNOVER® Lifecycle Manager job to run later.

More about running forms

Error-check

The Error-checking default is to do it in batch. You must do the pre-run error-check. Only an administrator can bypass this error-check step. You can do error checking interactively if you have a short form. Any errors detected are displayed in a window on this panel. Errors and warnings are reported. If you only have warnings and they are acceptable, you can submit the form. See the *TURNOVER® Lifecycle Manager User Guide*.

Submit form

Forms can be submitted immediately, on hold, or as a timed-job.

To specify a date and time, select ... **as timed job**, and then press Enter.

TURNOVER® Lifecycle Manager submits a job called `TOnnnnnnnnG`, where `nnnnnnnn` is your form number. Depending on how your application is set up, the `TOnnnnnnnnG` job remains on hold until the scheduler or an operator releases it, or it runs immediately.

If your application has been set up to submit the finalization phase, another job is submitted towards the end of the promotion process with a job name of `TOnnnnnnnnF`. For more information about the promotion process, see *[‘What happens when a form runs?’ on page 59](#)*.

Work with form

Select this to view or update the TURNOVER® Lifecycle Manager form.

Check for checked-out objects not on form

TURNOVER® Lifecycle Manager checks to see if you have missed any objects on the form. If you have used the **Project task reference** to reserve or check out objects and, subsequently, forgot to include them on the form, TURNOVER® Lifecycle Manager displays them. You can add them to the form, if appropriate.

Work with form distribution defaults

To use this option, you must have authority to maintain distribution defaults and distribution must be specified in the application level definition.

Log files

When a form runs, it produces two TURNOVER® Lifecycle Manager log reports and an IBM i job log.

What happens when a form runs?

A TURNOVER® Lifecycle Manager form runs in several phases.

1: Pre-run processing

Preliminary processing, including pre-run exits, pre-run commands, creation of working libraries, and so on.

You can specify pre-run commands to be run during phase 1 (`*PREFORM`) or just before an associated object is promoted (`*PRELINE`).

You can specify post-run commands to be run immediately after the object is promoted (`*POSTLINE`) or during phase 3 (`POSTFORM`).

2: Promotion processing

Source is copied, objects created or moved, folder objects promoted, data copied, logical files rebuilt (after physical file data is copied to the new file).

If more than one form is being run in a group, all forms are processed through phase 2, prior to starting phase 3.

3: Post-run processing

All post-run processing, including file object explodes (if you use Explode libraries), post-run commands, post-run exits, and so on. Once this phase is complete and before the Finalization phase begins, the form status changes to `FIN-OK` (Finalization OK). Whether finalization is done immediately or later in another job, your users can be allowed to use the objects once the form has reached `FIN-OK` status. If distribution is being done, a distribution job is submitted at this time.

If more than one form is being run in a group, all forms are processed through phase 3, prior to starting phase 4.

4: Finalization

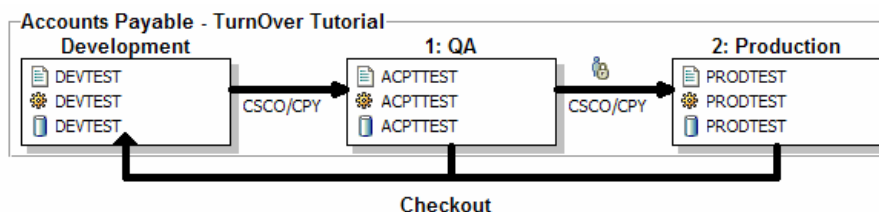
All closing processes, including source archiving, cross-reference rebuilds, deletion of from-source and objects, deletion of working libraries (if **Archive objects** is No).

TURNOVER® Lifecycle Manager can back out your changes if a compile-error is detected during promotion processing, and it can also undo your changes after the fact. It can also restart a promotion job, if, for example, the power fails or if someone inadvertently cancels your promotion job outside of TURNOVER® Lifecycle Manager's control.

For more information about what TURNOVER® Lifecycle Manager is doing during form processing, see 'Change process' the *TURNOVER® Lifecycle Manager Administrator's Guide*.

Testing and acceptance

If your company uses more than one level, as in the sample AP application, you might need to correct a problem discovered at an intermediate level such as acceptance testing (ACPTTEST in the diagram):



The normal procedure for this is to check out the object from that level (the level one ACPTTEST library in our sample schematic), correct the error, and then promote the test source and object back to that test level to complete your testing.

To checkout from your Worklist panel

- 1 Select the object at level one with option 21 (Checkout).

- 2 If there are only a few objects to fix, you can perform checkout interactively by pressing F19; otherwise, submit a checkout job to batch by pressing Enter.

The functions of the Enter key and F19 can be reversed by pressing F18 and setting a session default.

Promoting to a production environment

In a multi-level environment, your TURNOVER® Lifecycle Manager administrator is often responsible for scheduling and promoting your changes back into production. However, you might be required to do so yourself.

To prepare a TURNOVER® Lifecycle Manager form to promote to production (or any higher level)

- 1 Copy the level 1 form to create the level 2 form.
- 2 On your Worklist panel, select any object with option 43 (Copy form).
- 3 Press Enter or F19 (Submit to batch). TURNOVER® Lifecycle Manager creates a new form.

Depending on the target application definition and the contents of the form you are copying, TURNOVER® Lifecycle Manager includes additional parameters to control what is to be copied on the form. These include:

- *Copy commands:* any pre- or post-run commands on the (from) form are copied.
- *Copy form comments:* all comments on the (from) form are copied.
- *Check cross reference:* TURNOVER® Lifecycle Manager checks the cross-reference database and adds any related objects not already on the form to it for recompile.
- *Check related applications:* TURNOVER® Lifecycle Manager checks for objects on the form that exist in related applications and creates additional forms to recompile these objects.

- 4 Once you have created the next level form, **Nxt Opt** is either 47, in which case the form can be submitted to run (if you are authorized) or **Nxt Opt** is 48.
- 5 If the form status of the new form is **APL-PN** (Approval Pending), then you need to have the form approved first. Select any item on your worklist with **Nxt Opt** of 48, and then follow the instructions on the Form Approval panel.
- 6 When the form has been approved, it can be submitted to run.

For more information about what happens when a form runs, see [‘What happens when a form runs?’ on page 59](#)

Distributing to production computers

When your form runs successfully (form status is `FIN-OK` or `RAN-OK`), then, if you are running production on more than one system, the form is distributed automatically to your production systems.

If the form does not run successfully (form status `xxx-ERR` or `RECOVER`), or if the form runs but warning messages are detected, then the form is not distributed automatically. You may be asked to check the form. Once you are satisfied that everything is correct, your administrator can manually distribute the form.

Depending on how your application is managed, you may distribute source, objects, or both to remote locations. These can be controlled by system name, by application and by object type within application. For example, you may send source to system 020 but source and objects to system 030. You may send source for logical files, but objects for all other object types related to your application.

Distributions can occur at any application level. For example, you may want to distribute objects to a test computer prior to promoting to production.

Distributions to some remote systems may be via network connections; others by media such as tape.

Distribution of changes to remote computers is usually the responsibility of the TURNOVER® Lifecycle Manager administrator.

For more information about the distribution facilities, see the *TURNOVER® Lifecycle Manager User Guide*.

Updating the project task

To update your project task details or task status

- 1 Select any item in your worklist with option 25.

The Change a Task panel appears.

MM/DD/YY 10:00:15		Change a Task		Your Company, Inc.	
Project: APPR TURNOVER v101 Problems					
Task		0005			
Description		Chg credit-term verify program			
Resource		FISHER		F4=List	Jim Fisher
Status		D DONE		F4=List	
Requester		MUELLER		F4=List	John Mueller
Application		AP		F4=List	
Release					
Version					
Priority		3		1 to 3	
Priority sequence					
Duration		Planned:		Actual:	
Start date		Planned:	0/00/00	Actual:	0/00/00
End date		Planned:	0/00/00	Actual:	8/16/20
Date needed		0/00/00			
Date promised		0/00/00			
Task type		A ASAP		F4=List	
More...					
F3=Exit F4=List F5=Edit details F6=Add subtask F7=Requester comments					
F10=Additional parameters F12=Cancel F21=System command					

2 Update the task details or status (F4) or press F5 to view task details and append information.

Updating the task status automatically

You can set up TURNOVER® Lifecycle Manager to update the task status automatically when you check out source; when you run a TURNOVER® Lifecycle Manager form; when you distribute changes and, again, when you receive confirmations from your remote computers.

This capability greatly increases the integration between TURNOVER® Lifecycle Manager's change management and project management subsystems and reduces the need to update the task manually.

If you use intermediate test levels, you can set up TURNOVER® Lifecycle Manager to update task status to IN-QA when it promotes your changes to test, and to REVISIONS, if you check out of test.

For more information about how tasks status is updated automatically, see the *TURNOVER® Lifecycle Manager User Guide*.

For more information about changing tasks, see the *TURNOVER® Lifecycle Manager User Guide*.

Recording task time

If you account for your time, you can record and report your time in TURNOVER® Lifecycle Manager Projects.

To record your time

1 Press F22 on the worklist.

The Work with Programmer Worklist panel appears.

```

MM/DD/YY  10:00:15      Work with Programmer Worklist      Your Company, Inc.

Worklist  . . . . APPR0004      Chg credit-term verify program
Position to . . . . _____ (Object)      Apply filters . . . N (F17=Filters)

2=Chan
21=Chec

                                Add Time Sheet Entry

Resource . . . . FISHER      Jim Fisher

Obj
___ APP      Date . . . . . 8/13/20      Thursday August 13, 2020
___ APP      Project . . . . APPR 0004      F4=List
___ APP      Category . . . . -            F4=List
___          Comment . . . . _____

From time . . . 00 : 00      HH:MM - 24-Hour Clock
To time . . . . 00 : 00      HH:MM - 24-Hour Clock
--or--
Total time . . . 00 : 00      HH:MM

Paramet      F3=Exit  F4=List  F12=Cancel
==> _____
F21=Print
    
```

To report time for a task other than the one associated with your worklist item, move to **Project**, and then press F4.

To view the list of valid code values, move to **Category**. and then press F4.

To access your daily timesheet

Type SOFTTURNER/TIMESHEET on a command line, and then press Enter. A command prompt appears. Press Enter again.

```

MM/DD/YY      Work with Time Sheets      Your Company, Inc.
12:45:15      YOURSYS

Resource . . . FISHER      Jim Fisher
Date . . . . . 8/13/20      Thursday August 13, 2020

1=Add Time Entry  2=Change Time Entry  4=Delete Time Entry

Sub Cat.  -----Time-----
Project Task Task Code  From - To Total Comment
1  APPR  0001      P    8:00 17:00  9:00 Phew!_____

Total time for day . . . . . 9:00

F3=Exit F4=List  F6=Add Time Entry  F12=Cancel  F17=Previous day  F18=Next day
    
```

This panel display all of the time you have entered for a particular day. Press F17 to see previous days; press F18 to see later days.

Some companies have set up administrative projects with tasks such as holidays, meetings, sick time, and so on. Ask your TURNOVER® Lifecycle Manager administrator if this applies to your company.

To view the time reported by all resources for any task or project, select option 9 (View) time on the Work with Projects panel.

To use the timesheet reports, use the Reporting menu. For more information, see the *TURNOVER® Lifecycle Manager User Guide*.

4 Advanced topics

See

'More on the Programmer Worklist Manager' on page 65

'Setting user-defined PWM options' on page 66

'Setting TURNOVER® Lifecycle Manager user-defined options in PDM' on page 67

'Emergency changes' on page 68

'Using the multiple checkout feature' on page 70

'Resolving concurrent changes' on page 70

'TURNOVER® Lifecycle Manager commands' on page 71

'APIs, exits, overrides' on page 75

'Cross-referencing' on page 78

'Using the source compare utility' on page 80

'Using the source merge utility' on page 80

'Managing vendor software changes' on page 80

'Application versioning' on page 81

'Object versioning' on page 82

'Message IDs, message files, menus, and other object types' on page 82

'Managing ILE programs' on page 84

More on the Programmer Worklist Manager

To see a list of programmer worklists with which you are working or have worked, type `STRPWM` on a command line.

The Work with Programmer Worklists panel appears.

```

MM/DD/YY 11:15:22      Work with Programmer Worklists      Your Company, Inc.

Position to Worklist . . . . 
Filter by Programmer . . . . FISHER      Filter by Task status . . . 
Filter by Appl rel ver . . . 

1=Select  2=Change  3=Copy  4=Delete  5=Display task  6=Work with forms
7=Print   8=Clear Worklist Filters  9=Reassign Worklist

  Worklist  Description                      Status  App Rel Ver  Programmer
_ APPR0002  Add field to vendor file          DONE   AP           FISHER
_ APPR0004  Chg credit-term verify program    STARTED AP           FISHER

F3=Exit  F4=List  F5=Refresh  F6=Add  F7=Previous Worklists  F24=More keys
  
```

You can have several worklists active at a time; each one containing objects and source related to a different project task.

To add a worklist, press F6.

If you know the worklist with which you want to work, prompt the STRPWM command, and then specify the worklist name. This command is option 12 on the TURNOVER® Lifecycle Manager main menu.

To work with the last one with which you worked, type: STRPWM *PRV.

Setting user-defined PWM options

To view or change your user options, press F16 on the Programmer Worklist panel.

```

MM/DD/YY      Maintain User Defined Options      Your Company, Inc.
10:18:05      For: Programmer Worklist Manager

Protected options already exist as TURNOVER options.

File: TUSROPTF
Library: *LIBL

2=Change option  4=Delete option

Opt Description      Command
  2 Change
  3 Copy
  4 Delete
AP Abstract/Probe+  GO APLUS/APLUS
DC Display CMD      DSPCMD CMD("&OL"/"&OB")
DD Display Data      DSPPFM FILE("&OL"/"&OB")
DF Display File D    DSPFD FILE("&OL"/"&OB")
DP Display Program   DSPFGM PGM("&OL"/"&OB")
HK Hawkeye          HAWKEYE/HAWKEYE
PM Print member      STRSEU SRCFILE("&PL"/"&PF") SRCMBR("&PM") OPTION(6)
RC Rmv merge cmts    RMVMRGCMT FILE("&PL"/"&PF") MBR("&PM")
RM Remove Member     RMVM FILE("&PL"/"&PF") MBR("&PM")

F3=Exit  F6=Add option  F11=View 2  F12=Cancel
  
```

To change the options, you must have authority to the user option file.

The numeric options 1-50 and 99 are reserved by TURNOVER® Lifecycle Manager. For consistency, all TURNOVER® Lifecycle Manager users share this option file.

To customize your own set of user options, copy the user option file, TUSROPTF, to your own library and then change it.

To set your worklist to point to your own user-option file, press F18 on the worklist, press Page Down, and then change the user option file and library names.

TURNOVER® Lifecycle Manager is shipped with some user options, including:

Option	Description
DF	Display file description
DC	Display command
DD	Display data
RC	Remove merge comments that have been added by the TURNOVER® Lifecycle Manager merge command.

Using the available substitution variables, you can define your own user options.

Setting TURNOVER® Lifecycle Manager user-defined options in PDM

To maintain user-defined options, select option 9 on the PDM menu. Select or type the file and library, and then press Enter.

The Work with User-Defined Options panel appears.

Work with User-Defined Options				YOURSYS
File	:	QAU00PT	Member	: QAU00PT
Library	:	QGPL	Position to	: _____
Type options, press Enter.				
2=Change		3=Copy	4=Delete	5=Display
Opt	Option	Command		
—	AO	CHGOBJOWN OBJ(&L/&N) OBJTYPE(&T) NEWOWN(ADTOWNER)		
—	AP	ADDPRLIBL PRJ(&ZP) GRP(&ZG) SCAN(&ZH) SCHPTH(&ZS)		
—	C	CALL &O/&N		
—	CC	CHGCURLIB CURLIB(&L)		
—	CD	STRDFU OPTION(2)		
—	CK	DSPFD FILE(&L/&N) TYPE(*RCDFMT) OUTPUT(*PRINT)		
—	CL	CHGCURLIB CURLIB(&N)		
—	CM	STRSDA OPTION(2) SRCFILE(&L/&F) ??SRCMBR()		
—	CO	CHGOBJOWN OBJ(&L/&N) OBJTYPE(&T) NEWOWN(QPGMR) CUROWNNAUT(*SAME)		
—		More...		
Command				
====> _____				
F3=Exit		F4=Prompt	F5=Refresh	F6=Create
F9=Retrieve		F10=Command entry		F24=More keys

If you use PWM, you will use PDM less. The option you will use most often is AI. All other functions can be performed better from PWM.

TURNOVER® Lifecycle Manager user-defined options that you can copy to your own PDM user option file are:

Option	Description
AI	Add item to your worklist while working on the PDM Work with Members or Work with Objects panels. This assumes that you selected F20 from your PWM panel.
TP	Work with Project panel
T1	View or change application defaults
T4	Work with object history
T5	Checkout menu
T6	Work with TURNOVER® Lifecycle Manager forms
VC	Display or change checked out objects and source

If you copy these to your own PDM option file, you can change the option code, if you need to.

Emergency changes

There are two ways of performing emergency software changes in TURNOVER® Lifecycle Manager:

- emergency change application (recommended)

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.

- multiple checkout feature

The multiple checkout feature enables two programmers check out the same object at the same time, without defining a separate application.

TURNOVER® Lifecycle Manager processes and tracks emergency changes in the same way as it does normal changes.

Using an emergency change application

You can define one or more application definitions as an emergency change application. These enable you to check out TURNOVER® Lifecycle Manager source that may be checked out to another user. This method enables you to make changes and test them in a separate development library, before returning them to production.

To define an application as an emergency change application

On the Work with Application Definition panel, change the **Emergency** flag to ☒. For more information, see the *TURNOVER® Lifecycle Manager User Guide*.

An application that is defined as an emergency change application is the same as a 'normal' application, with one exception. When you need to make an emergency change using an emergency change application, you are allowed to check out source that has already been checked out to another programmer. In all other respects, works as any other application normally works. With TURNOVER® Lifecycle Manager's source-compare utility, 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 special messages to the programmer who currently has source checked out, to the application 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 indicate that emergency changes are being processed.

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 the 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). TURNOVER® Lifecycle Manager generates source compare output when the form runs.

Using the multiple checkout feature

The multiple checkout feature enables multiple users to check out the same object.

To set up multiple checkout feature

- 1 Select option 8, then option 5 on the TURNOVER® Lifecycle Manager main menu.

The Set Global TURNOVER Defaults (SETGLOBE) panel appears.

- 2 Set **Allow group checkout** to `Y`.

Using multiple checkout

When **Allow group checkout** is enabled, 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:

- `*YES` prevents other users from checking out the same object.
- `*NO` allows others to check out an object while you are working on it.

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, 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, TURNOVER® Lifecycle Manager displays a list of 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.

TURNOVER® Lifecycle Manager 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, then 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 you have checked out now that were changed concurrently by another programmer

- 1 On a command line, type `CHKCHGOBJ` (Check Changed Objects), 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 Reference	Approved
_ SLSF001	SLSLIB	RPG	ABFORTIS	10431	08/12/20	SLPR0321	PSMITH

F3=Exit F5=Refresh F12=Cancel F14=Display all

- 2 To confirm that the changes made by the other programmer have been made to your own version of the object, select option 1 (Approve).

Notes

- This command runs 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 also runs 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. Keep in mind that the multiple checkout rules apply to any level.
- Whenever you are not sure whether your version of source matches another version of source, use TURNOVER® Lifecycle Manager's source compare command, CMPSRCMBR or select the **Compare source** option on the Programmer Worklist.

TURNOVER® Lifecycle Manager commands

Commands for developers

All of these commands are available as options on the Worklist or other TURNOVER® Lifecycle Manager panels.

Command	Description
ADDPWLITM	Add worklist item
ADDTIMSHT	Add timesheet entry
CHGPWL	Change programmer worklist
CHGPWLITM	Change worklist item
CHKCHGOBJ	Check changed objects
CHKINMBR	Check-in by member
CHKINOBJ	Check-in objects
CHKOUTDFU	Checkout S/36 DFU
CHKOUTDOC	Checkout document
CHKOUTMEN	Checkout S/36 menu
CMPSRCMBR	Compare source member
CPYPWL	Copy programmer worklist
CPYPWLITM	Copy worklist item
CPYTSTDAT	Copy test data as post-run command
CRTPWL	Create worklist command
DISTDATA	Distribute data to remote systems
DLTPWL	Delete programmer worklist command
DLTPWLITM	Delete worklist item
DSPCHKOUT	Display checked out members command
DSPCOHST	Display checkout history
FASTTASK	FASTTASK project entry
HELPDESK	Helpdesk: Work with users
MOVPWLITM	Move worklist item
MGRSRCMBR	Merge source members
STRPWM	Start the programmer worklist manager

Command	Description
TADJPWLITM	Adjust worklist item
TCMPMSGF	Compare message file
TDSPOBJD	Display object description
TIMESHEET	Enter timesheet
TOPACK	Archive pack source member
TOPACKF	Archive pack source file
TOUNPACK	Archive unpack source member
TURNOVER	Main menu
VIEWDFTS	View application defaults command
VIEWGLOBE	View global defaults
VIEWXREF	View cross-reference
WORKTASK	Helpdesk: For reviewing incoming problems and request
WRKHLPTREE	Wisedesk: Work with help tree (search)

Report commands

Command	Description
LISTDFTS	List system default values report
LISTFORMS	List forms report
PRJCSTRPT	Project cost report
PRJDFNRPT	Project definition report
PRJPTYRPT	Project report by resource and priority
PRJRSRTIM	Project resource time report
PRJTSTRPT	Project task report
PRTTURFORM	Print form
REFACTRPT	Project reference report
RPTFORMDST	Form distribution report

Command	Description
RPTOBJDST	Object distribution report
RPTOBJHST	Object history report
RPTPGMRHST	Project programmer history report
RPTPGMRTIM	Project programming time report
RPTSRCUSE	Project source checkout report
TIMERPT	Project timesheet report

Setup and administration commands

Command	Description
AUDITADJ	Audit adjustment
AUDITRPT	Audit report
CHGTODDMF	Change distribution DDM files
CRTAPPDFN	Create application definition
CRTDEMOENV	Create demo environment
CRTFTFIL	Create project DDM files
CUSTFKEY	Set custom Fkey (F15) on PWM panel
DSPRELTO	Display current release level command
DSTFOR	Distribute forms
DUPREFOBJ	Duplicate reference object
DUPTSTOBJ	Reset DEVTEST command
LISTAUTH	List authorized users
LISTPGMRS	List programmer profiles
PRGFOR	Purge command
RCVFOR	Receive forms
RESETFOR	Reset form status
RGZTOFILES	Reorganize files

Command	Description
RNMAPP	Rename application definition
RSTARCSRC	Restore archived source command
SAVARCO	Save archived objects command
SAVARCS	Off-line archiving command
SAVARSBM	Submit archived object save
SETAUTCD	Set (product) authorization code
SETGLOBE	Set global defaults
TWRKCMSCH	Work with change management schemes
VIEWPTFHST	View PTF history file
WORKTODATA	Work with *DATA type objects
WRKDSTSYS	Work with distribution systems
WRKTOAPP	Work with applications
WRKTOENV	Work with environment file
WRKTOTYP	Work with type codes

APIs, exits, overrides

Application Program Interfaces (APIs)

All TURNOVER® Lifecycle Manager Application Program Interfaces (APIs) are in the form of commands. These can be included in programs to do a specific thing such as to create a form; add lines to a form, and so on. There are also interactive versions of some of these commands that you might find useful as shortcuts that can be run from a Worklist or PDM command line. The commands of greatest interest to the developer are highlighted.

Command	Description
ADDCMD	Add a command
ADDCMDI	Add a command (interactive)
ADDFORM	Add form

Command	Description
ADDFORMI	Add form (interactive)
ADDLINDLO	Add a DLO line to a form
ADDLINDLOI	Add a DLO line to a form (interactive)
ADDLINE	Add a line
ADDLINEI	Add a line (interactive)
APVFORM	Approve form command
CHECKIN	Manual check-in
CHECKOUT	Checkout
CHKOUTDLO	Checkout DLO objects
CPYFORM	Copy form command
CPYFORMI	Copy form command (interactive)
CRTRCVFRM	Create recovery form
CRTRCVFRMI	Create recovery form (interactive)
DSTFORM	Distribute form
DSTFORMI	Distribute form (interactive)
PRMDISFORM	Promote distributed forms
PRMFORM	Promote form
RCVFORM	Receive form
RUNFORM	Run form
SEQFORM	Re-sequence form

Exit programs

TURNOVER® Lifecycle Manager contains many program exit points.

For example, you can write a program that runs just before your TURNOVER® Lifecycle Manager form starts to run or during the final stages of the TURNOVER® Lifecycle Manager run.

For a list of exit points, select main menu option 8, and then option 7.

For more information about the program exit points, see the exits' online help and "Exits and APIs" in the *TURNOVER® Lifecycle Manager User Guide*.

Overrides

There are two ways to override normal processing that occurs when a TURNOVER® Lifecycle Manager form promotion job runs:

- code a pre-run or post-run command
- override the create commands used for any object you are promoting.

Pre-run and post-run commands

Pre-run commands can be run either during TURNOVER® Lifecycle Manager form initiation in Phase I or just before the object with which the command is associated is promoted.

Post-run commands can be run either immediately after the objects with which the command is associated is promoted or during the third phase of promotion processing, after all objects are promoted (but before the finalization phase).

There are two ways to enter a pre/post-run command:

- use option 38 on the programmer worklist
- edit the form as described below.

To enter a pre-run command on the form, edit the TURNOVER® Lifecycle Manager form.

To associate the command with an object on the form, select that object with option 11=PreRunCmd or option 12=PostRunCmd. For example, you might need to override a database file (OVRDBF) to successfully compile a program. When you promote that object in the future, TURNOVER® Lifecycle Manager prompts you to include the same pre-run or post-run command on the form.

You can condition a pre-run command so that, if the command fails, the job stops running (C) or continues to run (G). If a post-run command fails, TURNOVER® Lifecycle Manager does *not* roll back the form.

- C does not run subsequent commands.
- G continues with subsequent commands.

Note If you use the *explode data library* feature, you can substitute any library name parameter on a Post-run command with &ALLTGT; the all target library parameter. Then when the post-run command is run, TURNOVER® Lifecycle Manager substitutes the *Target Data Object library* names for the &ALLTGT parameter value and runs the post-run command repeatedly for each library in your explode data library list.

You might not be authorized to run certain commands. If you use a pre- or post-run command, see your TURNOVER® Lifecycle Manager administrator to be sure the command you plan to use will run.

More information

For more information about pre- and post-run commands, see the *TURNOVER® Lifecycle Manager User Guide*.

Overriding create commands

You can override the create command that TURNOVER® Lifecycle Manager uses when it creates an object by viewing the TURNOVER® Lifecycle Manager Form Maintenance panel and selecting that object with Option 10=CreateParms. TURNOVER® Lifecycle Manager prompts the command, and you can then change whatever you need to change. For example, you might need to change the file size parameter for a new physical file.

TURNOVER® Lifecycle Manager maintains its own object type attribute file and create commands for each object type (where appropriate). You can change command defaults used by TURNOVER® Lifecycle Manager without affecting the IBM i command defaults. To see these commands and command defaults, select main menu option 8, and then select option 4.

When you override a create command, TURNOVER® Lifecycle Manager remembers the override and displays it the next time that object is compiled in development using worklist option 36, or is added to a form.

Project user-defined field exit programs

In TURNOVER® Lifecycle Manager Projects, you can define user-defined fields; you can assign up to ten of these fields to any project. The fields appear on Add and Change a Task and Task filter panels. You can define a list of valid values for each user-defined field. You can condition the contents of a field on the contents of another field by specifying a parent/child relationship between the two fields when defining the fields and their valid values. TURNOVER® Lifecycle Manager displays the list both during data entry, when you put the cursor in the field and press F4, and again when verifying the value entered. Alternatively, you can define your own exit programs to display a list of field values or to verify the values entered in a field.

For more information, see the *TURNOVER® Lifecycle Manager User Guide*.

Cross-referencing

TURNOVER® Lifecycle Manager uses its own internal cross-reference utility, or you can use Hawkeye's PATHFiNDER or HelpSystems's Abstract. You can also use TURNOVER® Lifecycle Manager and either PATHFiNDER or Abstract at the same time. 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.

There are several points of interface between TURNOVER® Lifecycle Manager and PATHFiNDER or Abstract.

- When you set up your application, you can select an option to build a cross-reference file. You need to do this only once. You also set your application defaults to indicate you are using Pathfinder, Abstract or TURNOVER® Lifecycle Manager's cross-reference utility. This can vary by application and level; that is, you could use Pathfinder for one application, TURNOVER® Lifecycle Manager's cross-reference for another application and Abstract for another.
- During analysis you can view the cross-reference information for any object or member and add them to your worklist (see '[Checking cross-references' on page 32](#)) or to a form. You can also check out related objects at the same time you add them to your worklist.
- When you build a form to promote your objects, TURNOVER® Lifecycle Manager displays related objects that are not already on the form. You can add these related objects to the form so that they are recompiled. This can be done automatically. Even if a file is used by an object in a different application, TURNOVER® Lifecycle Manager shows you the object and allows you to build a form to recreate the related objects.
- When the form runs, TURNOVER® Lifecycle Manager updates its cross-reference database so you never have to rebuild it. For PATHFiNDER and Abstract, you can refresh your cross-reference database dynamically (or as a scheduled event, as you do now). Before you decide to rebuild every time a form runs, you must consider the affect on system performance.
- Synchronizer, the vendor software merge system, also uses the cross-referencing database. For more information, see the *TURNOVER® Lifecycle Manager Synchronizer User Guide*.
- If you have a multi-level environment and promote a file object to test (or any intermediate level), you can rebuild dependent objects in test from production source available at a higher level. This means that you can be sure you have not adversely affected related objects, even if you have not actually changed them.
- In PATHFiNDER and Abstract, you can set up user-defined options which enable you to add objects to your Programmer worklist or check out any objects you need to work on as you view the product's cross-reference database. Also, you can access *where used* information while viewing the Work with Object History panel.

Advantages and disadvantages of using cross-reference in TURNOVER® Lifecycle Manager

TURNOVER® Lifecycle Manager's cross-reference is tightly integrated with the Programmer worklist. Cross-reference rebuilds are done every time a TURNOVER® Lifecycle Manager form runs. If you choose to have TURNOVER® Lifecycle Manager maintain your cross-reference in this way, then the cross-reference database will always be up-to-date.

However, there are some relationships that PATHFiNDER and Abstract identify which TURNOVER® Lifecycle Manager does not; for example, if you use variable names in your program calls or if you need field-level cross-referencing.

Using the source compare utility

There are several points during the analysis and change process when it makes sense to run a source compare program. You can use either TURNOVER® Lifecycle Manager's own source compare program or another one. For more information about setup, see the *TURNOVER® Lifecycle Manager User Guide*.

The source compare utility can be used in the following ways:

- A user-defined option (SC) on the TURNOVER® Lifecycle Manager form Maintenance panel. TURNOVER® Lifecycle Manager compares the From and Target Source Member for the lines you select and prints the source compare report of differences (of what changed).
- A user-defined option (SC) on the Work with TURNOVER® Lifecycle Manager forms panel. TURNOVER® Lifecycle Manager submits a batch job to produce source compare reports for all objects on the TURNOVER® Lifecycle Manager form.
- Option 12 on the Work with Object History panel. TURNOVER® Lifecycle Manager prints a source compare report for any two source members you select.
- Option 12 on the Work with Programmer Worklist panel. TURNOVER® Lifecycle Manager prints or displays the source compare report for the members you select.

Synchronizer also uses the source compare and source merge utility in several places.

The source compare utility command, `CMPSRCMBR` in the `SOFTTURN` library, can be run from any command line.

Using the source merge utility

- 1 Type `MRGSRMBR` on a command line.
- 2 Press F4, and then fill in the parameter values.

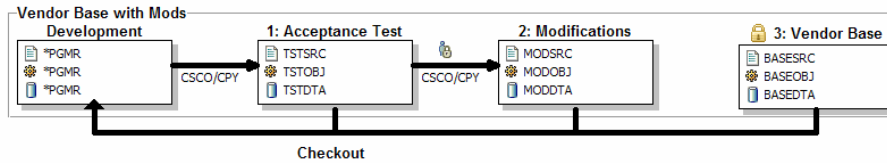
For more information about the source merge utility, see the *TURNOVER® Lifecycle Manager Synchronizer User Guide*.

Managing vendor software changes

The ability to manage objects and check out source in a vendor environment is especially useful if you are modifying software and want to preserve the vendor's base versions intact. This approach can also be used to manage the development of a new release of your own software.

TURNOVER® Lifecycle Manager enables you to specify one or more application levels (three levels are defined in the diagram below) to define the base libraries and source file names for your vendor's software.

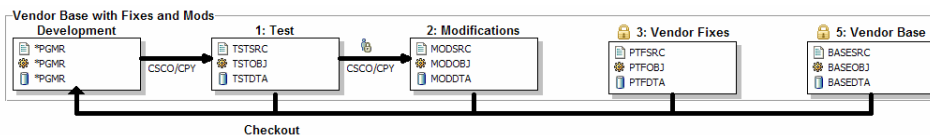
Sample application for vendor software



In the example, this level is *permanently locked* (controlled by a field on the **Application Definition** entry). Most likely, you will never turn objects over to that level (when you update the base with a new release from the vendor; but by using a different application definition for that purpose.)

When you check out source for an object, TURNOVER® Lifecycle Manager looks for the object and source by referring to the highest-level application definition *that is not permanently locked* (**Mod Src** library in diagram below). If it does not find the source there (for example, if the base version had never been modified before), TURNOVER® Lifecycle Manager checks for the source in the locked (base) level (level 3). If the source is found, it is copied to your development library. TURNOVER® Lifecycle Manager creates a checkout record for the source member as if it had been found in the modification source file. Actually, it reserves the name at that level. When you create a form to promote the object, the object is added to the modification level libraries.

If you have more than one permanently locked level such as a fix library (level 3) and a base library (level 4), TURNOVER® Lifecycle Manager searches the fix library and then the base library:



The Base level does not have to contain software from a vendor, but may be the previous release of an application you have developed. The modification level may represent work in progress for a future release.

For more information about defining permanently locked levels, see the *TURNOVER® Lifecycle Manager User Guide*.

Application versioning

Related application definitions enable you to track and maintain different versions of the same program in different version libraries. For example, you might have different versions of a program for different locations or companies that you support, or for different releases of your software, or both.

For more information, see 'Managing Multiple Versions' in the *TURNOVER® Lifecycle Manager Administrator's Guide*.

Object versioning

TURNOVER® Lifecycle Manager maintains internal version numbers for all objects it promotes. Optionally, it can stamp the object description with the version number and other control information.

TURNOVER® Lifecycle Manager provides its own Display Object Description (`TDSPOBJD`) command, which formats version stamping information for you.

For more information, see 'Managing Multiple Versions' in the *TURNOVER® Lifecycle Manager Administrator's Guide*.

Message IDs, message files, menus, and other object types

Message IDs

You can add individual message IDs to your worklist and then check them out, change them, and promote them when you are through.

- If a message file does not exist in your development library, TURNOVER® Lifecycle Manager creates one for you.
- If a message file does exist in development, TURNOVER® Lifecycle Manager adds the message ID to the message file, but it does not overlay a message if it already exist in your development message file.

Option 32 (Edit) on the worklist uses the `WRKMSGF` command rather than SEU or SDA.

For more information about checking out and promoting individual message IDs, see 'Handling message IDs' in the *TURNOVER® Lifecycle Manager Administrator's Guide*.

To delete message IDs

Check out your message file and duplicate the entire file into development. Change it using the `WRKMSGF` command; and delete and add the message file when you promote it so TURNOVER® Lifecycle Manager replaces the entire file in production.

Message files

When message files are promoted, they are always merged using the `MRGMSGF` command. If the message file is new, it is moved or duplicated into the target library. When you are changing or adding messages to a message file, you should check out and promote individual messages. However an alternate method is to check out the entire message file. This duplicates the message file from production into your development library. Include new or changed messages in a development message file; then promote the entire message file. When the message file is promoted, it is merged into your production file.

To compare two message files

Use the Compare Message File (TCMPMSGF) command.

Menus

TURNOVER® Lifecycle Manager enables you to manage IBM i menus.

- If you are reserving a menu name, TURNOVER® Lifecycle Manager reserves names for all the necessary object types that make up a menu.
- If you check out a menu, TURNOVER® Lifecycle Manager checks out all of the relevant menu components.

For more information, see 'Menus, DFUs, and queries' in the *TURNOVER® Lifecycle Manager Administrator's Guide*.

Non-source object types

Objects that are created directly by a command rather than from source, such as data areas and job queues, are always promoted using method MO (Move Object) or CD (Create Duplicate). How does TURNOVER® Lifecycle Manager determine whether an object is a non-source type? You can view TURNOVER® Lifecycle Manager's type file by selecting main menu option 8 and then option 4. Those types that have no commands associated with them, except for *DATA and *SRC types, are always moved or duplicated.

If, in the application line defaults, you set **Delete test object** to Y (Yes), the method for that object defaults to MO. If you set **Delete test object** to N (No), the method for that object defaults to CD. However, the method determined by settings in the application type codes defaults overrides the method specified in the line-level defaults.

Source-only object types

Source members such as S/36 OCL, RPG copybook code or documentation text, are referred to as "source only" types; they are promoted but *do not* create an IBM i object directly. When these are promoted, TURNOVER® Lifecycle Manager simply copies the source forward using method CS (Copy Source). View the TURNOVER® Lifecycle Manager type file by selecting main menu option 8 and then option 4. Those types designated *SRC, such as CPYRPG, are considered source-only types.

*DATA object types

*DATA object types manage the contents of files rather than the file itself. For example, you might have an application that uses control files that manage certain programming functions. Frequently, such control files are managed like other objects you need to change, except the object is data. The data changes, not the format of the file in which the data is stored.

For more information, see 'Working with *DATA definitions' in the *TURNOVER® Lifecycle Manager User Guide*.

Note You can use *change management schemes* to selectively override, for a user-defined object type, almost any change management method that you would execute for an object. You can use the sample change management schemes that came with this release as examples, or create your own schemes. For more information, see 'Managing User-Defined Object Types' in the *TURNOVER® Lifecycle Manager User Guide*.

Managing ILE programs

You can create and link (bind) Integrated Language Environment (ILE) modules into one or more programs or service programs. This makes program calls more efficient and the bound program more secure. (You can run only a program, not a module. Therefore, you have to secure only the program.) TURNOVER® Lifecycle Manager enables you to manage ILE modules and programs. All ILE language compilers and creation methods are supported.

TURNOVER® Lifecycle Manager includes these type definitions to support all of the various ILE entities:

Program	Description
BND	Bind (source-only type)
BNDDIR	Binding directory (object-only type)
CBLMOD	CBL module
CBLPGM	CBL ILE program
CBLSRV	CBL service program
CLMOD	CL module
CLPGM	CL ILE program
CLSRV	CL service program
CMOD	C module
CPGM	C ILE program
CSRV	C service program
RPGMOD	RPG module
RPGPGMT	RPG ILE program
RPGSRV	RPG service program

For more information about managing ILE programs, see 'Managing ILE Programs' in the *TURNOVER® Lifecycle Manager User Guide*.

Creating ILE modules

If you intend to create and maintain ILE `*MODULE` object types, you must use the appropriate type code for handling these object types. Generally, you use this method when a module object needs to be “bound” into several programs or service programs. The TURNOVER® Lifecycle Manager type code for creating modules uses the `CRTxxxMOD` command. You should create a separate library in your application definition for storing your `*MODULE` objects, but you can also use existing object libraries.

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 also supports its cross-reference database for `*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 ILE programs

To create an ILE program object, do one of the following:

- If the program contains more than one module: Use the `CRTPGM` command, and then specify the modules that make up the program.
- If the program contains a single module: 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.

Using CRTPGM to create programs

The `CRTPGM` command requires that you specify all of the modules and service programs that make up your program each time you want to recreate it, even if you have only updated a single module. To address this issue, the TURNOVER® Lifecycle Manager command, `TCRTPGM`, serves as a “wrapper” for the `CRTPGM` command. The `TCRTPGM` command uses operating system APIs to automatically 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.

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. Because TURNOVER® Lifecycle Manager relies on this information during checkout and auditing, this presented a problem in the past. In old releases, we created the `TCBPF` command to address this issue. In recent releases, we enhanced the way TURNOVER® Lifecycle Manager handles ILE bound programs, as is described later in this section.

Defining the underlying create command

Note 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	Change a Type Code	Your Company, Inc.
14:35:20		YOURSYS
Type code	RPGLE	Attribute RPGLE
Description	ILE RPG Program	Source file name . . QRPGLSRC
Object type	*PGM	Data object N
Sequence	500	
Create command . . .	CRTBNDRPG PGM("&LI"/"&OB") SRCFILE("&SL"/"&SF") SRCMBR("&SM") DBGVIEW("&U1") OPTIMIZE("&U2") TGTRLS("&TR")	
"&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" = Optimization level		"&U1" = Debugging views
"&U4" = Unassigned		"&U3" = Target Release
"&U6" = Unassigned		"&U5" = Unassigned
"&U8" = Unassigned		"&U7" = Unassigned
		"&U9" = Unassigned
F3=Exit	F4=Prompt command	F7=Select Applications F12=Cancel

You can also:

- use the TCBP command.
- use the Post-Form Exit (Exit 1) to have a program 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.

ILE service programs are language independent. You can use one type code for all languages. When setting up a service program type code, you must consider the *sequence number*. Service programs are created from modules, so they should have a higher sequence number than modules. They are also used to create programs, so 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 (you can also use MO or CD). The same is true for programs using the CRTPGM method.

Binding language source (BND)

This is a source-only type that contains statements that you can use when creating service programs. These are supported by TURNOVER® Lifecycle Manager.

Bind directory (BNDDIR)

This is an object-only type; that is, it is moved or duplicated from test to target libraries. You can check out a bind directory using worklist option 21. You can work with the directory and add directory entries to a directory 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.



www.softlanding.com

We welcome feedback on our documentation. Please email us at:
tech.authors@softlanding.com

www.unicomglobal.com