# TURNOVER® FOR ISERIES

## Application Planning Guide

UNICOM® Systems
A Division of UNICOM Global

## Publication number

TIAA-1010-00 (October 2018)

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

## Documentation set

The documentation relating to this product includes:

- Getting Started with TURNOVER® for iSeries Client
- Getting Started with TURNOVER® for iSeries
- TURNOVER® for iSeries User Guide
- TURNOVER® for iSeries Developer's Guide
- Application Planning Guide
- Managing IFS Objects with TURNOVER® for iSeries
- TURNOVER® for iSeries PWM Quick Reference
- Synchronizer User Guide

## Copyright notice



## TURNOVER® for iSeries



## Trademark acknowledgements

# Table of Contents

## Contents

# PREFACE

The information in this guide provides information to help you understand TURNOVER® for iSeries and TURNOVER® for iSeries application definitions.

Many procedures described in this guide are based on the TURNOVER® for iSeries Administrator module. We suggest that you install and configure that software; enroll yourself as a TURNOVER® for iSeries user; and give yourself authority for the TURNOVER® for iSeries application models named 01M – 09M before continuing with this document.

## WHO SHOULD USE THIS GUIDE

This guide is intended for Information Technology professionals and management personnel who are evaluating the TURNOVER® for iSeries product or planning its implementation in their company.

You, the reader of this guide, should have a general knowledge of the IBM i, be familiar with Security Administration, and understand the organization of your company's applications and databases. You should also have some understanding of, and experience with, Command Language (CL).

### Recommendations

Throughout this guide you will be given a complete explanation of TURNOVER® for iSeries features and procedures. Where appropriate, we have provided additional notes that contain our recommendations to help you with the decisions you need to make. We clearly label our recommendations like this:

> **UNICOM Systems, Inc. Recommends**
>
> Our first recommendation is probably the most important. We suggest that you assign someone to be your TURNOVER® for iSeries Administrator. This person does not need to be a programmer – in fact, most data processing auditors recommend that you assign someone outside of programming to act as your TURNOVER® for iSeries Administrator. This person might be an operations manager, quality assurance specialist or database administrator. Ideally, this person will understand the technical issues related to system security, data integrity, and object management, and will have some programming experience.

### Cautions

In every system there are choices – some choices may come back to haunt you later, especially where your production environment is concerned. We have provided cautions whenever we felt that you will need to make a particularly important choice.

Cautions look like this:

> **UNICOM Systems, Inc. Caution!**
>
> Although we have provided you with the capability to move objects (MO) and create duplicate objects (CD) from test libraries to production libraries, we recommend that you use the option CSCO (Copy Source to production and Compile Object) to ensure that the object is always created from the source member in your production library and that the correct files are used when creating the object.

## The TURNOVER® for iSeries Information Library

Other sources of information about TURNOVER® for iSeries include:

- The TURNOVER® for iSeries information library, consisting of this ***Getting Started with TURNOVER® for iSeries*** manual, a user guide, and a tutorial:
  - ***Getting Started with TURNOVER® for iSeries*** provides instructions for installing TURNOVER® for iSeries on your computer (for the first time), upgrading to a new release, and applying changes to an existing release.
  - The ***TURNOVER® for iSeries Tutorial*** presents an example of a TURNOVER® for iSeries application, and walks you through the steps you would typically perform in your actual working environments.
  - The ***TURNOVER® for iSeries Developer's Guide*** contains topics of specific interest to software developers. It also serves as an effective introduction to TURNOVER® for iSeries.
  - The ***Synchronizer User Guide*** describes all of the features of the Synchronizer system, including the source compare and merge programs.
- The ***TURNOVER® for iSeries Application Planning Guide*** provides information to help you understand TURNOVER® for iSeries and TURNOVER® for iSeries application definitions. (This information is user interface independent.)
- The TURNOVER® for iSeries Supplements and Technical Bulletins are articles on specialized or advanced topics.
- The TURNOVER® for iSeries Interface Guides describe how to use TURNOVER® for iSeries to manage objects created by other products.[1]

[1] For a complete list of products with which TURNOVER® for iSeries interfaces, visit the SoftLanding website or contact your UNICOM Systems, Inc. Sales Representative.

- The ***Getting Started with the TURNOVER® for iSeries Client*** guide provides installation and configuration instructions for using the Eclipse-based TURNOVER® for iSeries plug-ins, either in the standalone TURNOVER® for iSeries Client or within an Eclipse-based IDE such as IBM's RDi.

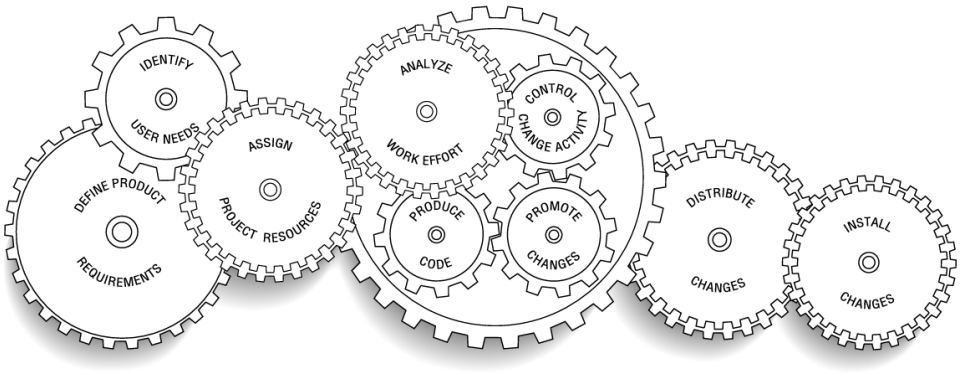
- ***TURNOVER® for iSeries Supplement #66: Implementing TURNOVER® for SVN*** provides complete information for installing and configuring the TURNOVER® for SVN version control product for your Web/PC/Java development projects. This product integrates seamlessly with the Eclipse-based TURNOVER® for iSeries plug-ins running either in the TURNOVER® for iSeries Client or another Eclipse-based IDE such as IBM's RDi.

In addition, Help text (both field sensitive and extended help) is available for every display. Position the cursor to any field, or to the panel title, and press the F1/Help key.

All TURNOVER® for iSeries documentation is available as Adobe™ Acrobat® PDF files within the TurnOverDocsClients.zip file. Using the Acrobat® Reader with Search (also provided), you can search, view, and print TURNOVER® for iSeries information.

# CHAPTER 1: OVERVIEW OF TURNOVER® FOR iSERIES

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




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

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

You decide which features to implement based on your needs. And, because TURNOVER® for iSeries' feature set is integrated into one package, it is easy to adapt your implementation as your requirements evolve.

# CHAPTER 1: OVERVIEW OF TURNOVER® FOR iSERIES

## UNICOM Systems, Inc. Recommends

This ***Overview*** describes how TURNOVER® for iSeries manages the change process. In addition to reading this ***Overview*** – and for a complete introduction to TURNOVER® for iSeries – we recommend that you read this guide entirely, along with the ***Overview*** sections in chapters **1**, **5**, **6**, **9**, and **11** in the ***TURNOVER® for iSeries User Guide***; the ***Introduction*** in the ***Developer's Guide***; and the ***Introduction*** in the ***Synchronizer User Guide***. These sections introduce you to the topics of planning and defining your applications, checking out source, the promotion request (TURNOVER® for iSeries form), TURNOVER® for iSeries Projects, distribution, and the Synchronizer system.

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

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

This document introduces you to change management concepts and the ways that TURNOVER® for iSeries helps you become a more effective **Change Agent**. It assumes that you are the person who will set up your software project tracking and development environments, and that you will also work as a developer.[2]

We refer to the person who sets up projects and applications and enrolls developers in TURNOVER® for iSeries' security files as the TURNOVER® for iSeries Administrator. In fact, s/he may be a manager, technical support programmer, operations manager, database administrator, or a person who fulfills a combination of any of those roles.

It might not be your responsibility to set up your environment. However, if you are going to be involved in decisions about how TURNOVER® for iSeries will be used at your company, then you should understand the TURNOVER® for iSeries system architecture. This way, you'll be able to contribute to the decisions made and to understand how TURNOVER® for iSeries can improve productivity at your company.

[2] If you will not be involved in setting up TURNOVER® for iSeries, but you will be working as a developer, you might prefer to read the *TURNOVER® for iSeries **Developer's Guide***.

# HOW TURNOVER® FOR iSERIES IMPROVES PRODUCTIVITY - EXAMPLES

These examples illustrate one or more features of TURNOVER® for iSeries and provide you with a quick synopsis of how you might approach a particular problem or perform a particular task.

Ask yourself how much time you would save by using the features illustrated. And don't forget to add in all of the time saved because you avoided an error, or because you could access change history that you did not have before.

## Example: Add a field to a physical file

As we all know, this activity requires recompiling all of the logical files and programs that refer to the physical file.

**Without TURNOVER® for iSeries**, you would have to:

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

**With TURNOVER® for iSeries**, you would:

1. Check out the PF source.
2. Change the source.
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.

## CHAPTER 1: OVERVIEW OF TURNOVER® FOR iSERIES

**TURNOVER® for iSeries** takes care of the rest – without making an error.

You want to undo your changes after they are installed?

**Without TURNOVER® for iSeries**, you would have to reverse all of the former steps, by hand, one at a time.

**With TURNOVER® for iSeries**, you simply build and run a recovery form. The rest is automatic.

### Example: Make an emergency change

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

**Without TURNOVER® for iSeries**, you would have to:

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

**With TURNOVER® for iSeries**, you would:

1. Check out the object using an emergency change application
2. Make the necessary change to the source in a development library
3. Run the emergency change form to promote the change and notify the programmer

**TURNOVER® for iSeries** automatically takes care of the rest -- including notifying Ed to remind him to make the same change.

## A few more simple examples

Sometimes we forget the fact that even though a situation is simple, it consumes time – especially if it occurs frequently!

- **S**uppose Tony needed to **check for an object lock** on the entry operator file before installing changes, every time one of several objects changed.

  **With TURNOVER® for iSeries**, you can be sure you do not pull the rug out from under active users.

- **S**uppose you need to change **a PF with 8 members**, all with live data.

  **With TURNOVER® for iSeries**, your physical files members will be preserved whenever you make a change to a file.

- **S**uppose you want to **change the same database file format twenty times** in twenty different company libraries.

  **With TURNOVER® for iSeries**, your change can ripple through a list of database libraries, effortlessly.

- **S**uppose you need to identify **what objects were changed when a problem was fixed** in January of 2002.

  **With TURNOVER® for iSeries**, you can easily retrieve a list of objects related to any historical task.

- **S**uppose you want to **verify all of the source and objects** before promoting them?

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

- **S**uppose you want to promote **200 programs** and related objects at 3 a.m., while still getting a good night's sleep.

  **With TURNOVER® for iSeries'** many verification checks and automatic rollback, your likelihood of success is greater -- AND if an error does occur, rollback is automatic.

Chances are you'd rather spend your time doing development instead of copying source, compiling objects, setting object authority, and fixing installation errors. With TURNOVER® for iSeries, your role ends when you build your promotion request **form**. A change administrator can take over from there, freeing you to go on to more development.

# CHAPTER 2: ABOUT APPLICATION DEFINITIONS

In this chapter, you will learn about the TURNOVER® for iSeries element that is the core of your TURNOVER® for iSeries workflow: application definitions.

## WHAT IS A SOFTWARE APPLICATION?

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

You might choose to manage these collectively and define only one application definition in TURNOVER® for iSeries for all of them, or you might manage each one separately. On the other hand, you might create a "DBA" application definition to manage all data object types, and one or more other application definitions to manage all non-data, program-related object types.

## WHAT IS A TURNOVER® FOR iSERIES APPLICATION DEFINITION?

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

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

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

## CHAPTER 2: ABOUT APPLICATION DEFINITIONS

# ABOUT APPLICATION DIAGRAMS

We find that an application diagram is a convenient way of helping you visualize what your TURNOVER® for iSeries application definition is set up to do. Once this mental image is clear, the rest is easy.

The diagram of the AP below was created using TURNOVER® for iSeries' new Application Designer. It shows a sample application schematic that we will refer to throughout this document. The setup reflected by this particular diagram is not the only choice you have! TURNOVER® for iSeries' greatest strength is its ability to manage software applications in many different ways.

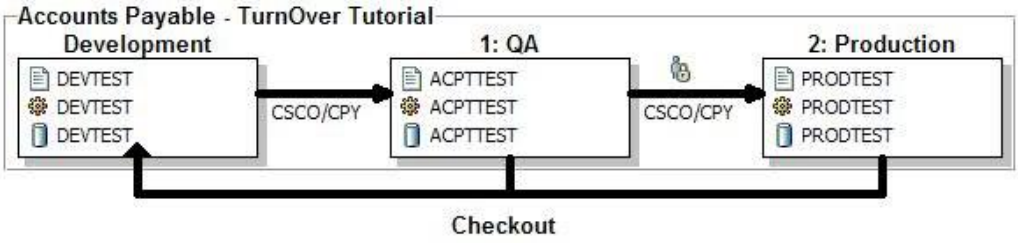



**Figure 1: Sample application diagram of application AP Release 0 Version 0**

## Understanding an application diagram

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

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

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

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

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

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

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

- Should source be archived when it is replaced? Where?
- Should your application definition enforce object naming rules?
- How do you want source code and objects handled during a promotion?
- Do you want to distribute objects to other computer(s) when you promote changes? Objects and source? Source only?
- Should someone in your organization approve changes (on-line) before they are implemented?
- What user profile should own the objects TURNOVER® for iSeries creates?
- Of what object types is your software application constructed?

This is just a sampling of the rules contained in a TURNOVER® for iSeries application definition.

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

Even if you do not have authority to define applications, you can view them.[3] We suggest that you browse an application definition to familiarize yourself with some of the defaults and rules. Online help is available in all TURNOVER® for iSeries user interfaces.

* For a complete discussion of application defaults, see the online help and ***Chapter 1: Working with Application Definitions*** in the ***TURNOVER® for iSeries User Guide***.

[3] There are a variety of ways to look at applications, depending on which user interface you are using. We recommend using the Eclipse-based TURNOVER® for iSeries Administrator module to manage your application definitions, and the descriptions in this book refer to that product.

## CHAPTER 2: ABOUT APPLICATION DEFINITIONS

# UNDERSTANDING THE TURNOVER® FOR iSERIES WORKFLOW

Once you have been assigned a task and have determined what objects you need to change or create (more on how TURNOVER® for iSeries supports these activities later), you will likely take the following actions. (To keep you from having to flip pages, we'll reproduce our diagram here.)

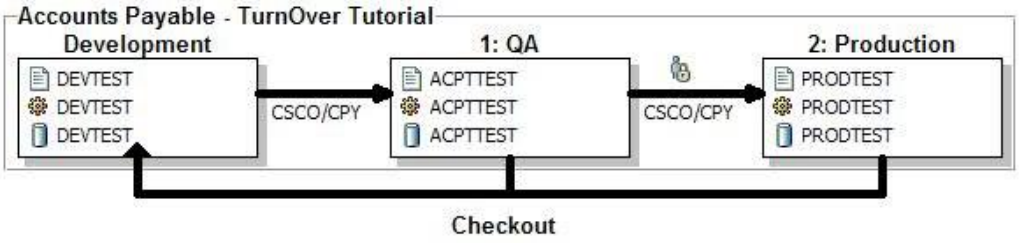



**Figure 2: Sample application diagram of application AP Release 0 Version 0**

1. **Check out objects and source from production**. TURNOVER® for iSeries copies the source from your production source library to your development library (non-source objects are duplicated). Depending on how your application is set up, TURNOVER® for iSeries will not allow others to check out the same objects. (For emergency changes, read about emergency application definitions in ***TURNOVER® for iSeries Supplement #24: Emergency Changes in TURNOVER® for iSeries***.) Using TURNOVER® for iSeries' cross-reference facilities, you can easily identify dependent objects, so you can feel confident that you know about all the objects you need to change or re-compile.

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

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

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

5. **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 system message or by email. Each user will review and approve your form. Also, if you want to, you can print a hard copy of the form and affix signatures to that.

6. **Run TURNOVER® for iSeries form**. Your TURNOVER® for iSeries Administrator will submit the form to run. Forms can be scheduled to run at a date and time you specify.

7. **The form runs**. During form processing, TURNOVER® for iSeries copies the source to the target library and compiles the object into the appropriate target object library(s). TURNOVER® for iSeries copies or moves non-source objects, as your application definition indicates, and duplicates source that you do not want compiled, such as RPG /COPY code or S/36 OCL. Old source and objects are archived.* TURNOVER® for iSeries handles the source and objects in the From library as your application indicates. If errors occur, TURNOVER® for iSeries rolls your changes back automatically.

8. **Check out from Acceptance (or any intermediate application level)**. If errors are detected during acceptance testing, you will check those objects out *of the acceptance test level*; then you revise them and promote them back into Acceptance for additional testing. This may happen several times.

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

10. **Run Production form**. Your Administrator submits the form to run. It can be scheduled to run at any time. When it does run, it will be rolled back if errors are detected.

11. **Back out form**. If after a form runs successfully you are not satisfied with the result, you can roll it back. You can roll back the entire form, or any subset of the objects thereon, if necessary.

12. **Distribute form**. A form is also a distribution request. If you are running a network of IBM i systems – whether physical systems or multiple partitions on a single system – TURNOVER® for iSeries can distribute a form after it runs successfully on the development computer. You can configure the schedule such that this happens immediately, or at some later date and time. When TURNOVER® for iSeries distributes a form, the objects bundled together with TURNOVER® for iSeries control information, and either written to tape or sent across your communications network.[4]

13. **Receive and run form**. On your remote production computer(s), your designated TURNOVER® for iSeries user receives forms and runs them just as they were run on the development computer. If you distribute from level 2 of your AP application definition, then you need a corresponding application definition for AP level 2 on the remote computer. That application definition controls how the objects are managed on *that* computer.

* Archiving of source or objects is optional.

[4] TURNOVER® for iSeries can distribute both native IBM i objects, and objects that reside in files systems on other platforms.

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

15. **Update form status**. Once objects have been distributed and installed on each remote computer, TURNOVER® for iSeries updates the status on the development computer and notifies people responsible for the change.

In the ***Developer's Guide***, each of the steps – from entering a change request through distributing a change to a remote computer – is illustrated in greater detail. That guide also introduces you to Helpdesk and Project management functions, which TURNOVER® for iSeries supports. For consistency, the schematic of the AP application is referenced throughout.

# CHAPTER 3: SETTING UP TURNOVER® FOR iSERIES

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

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

For a more in-depth discussion of TURNOVER® for iSeries and IBM i security, read ***TURNOVER® for iSeries Supplement #15: Security Recommendations***.

## ABOUT SECURITY

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

TURNOVER® for iSeries also assumes that passwords for user profiles that are the owners of your production objects (such as QSECOFR) are either not known by users other than the system manager, or that their passwords are set to ***NONE**. If programmers or operators can sign on as QSECOFR, or if they run procedures that operate under the authority of QSECOFR (a favorite trick of programmers to circumvent security), then they can still modify objects in production and bypass TURNOVER® for iSeries' procedures. If someone has ***ALLOBJ** authority, no application or system can prevent them from getting at production files and programs.

It is up to you to promote and enforce good security practices. If you do not, production libraries can become corrupted and, other than running TURNOVER® for iSeries' Audit report periodically, you have no reliable means of auditing to determine how a change was made.

This section provides schematics of several approaches you can take when setting up TURNOVER® for iSeries to model your operations. If you are new to the IBM i, it will acquaint you with some of the issues and help you decide how to set up your system. It covers a variety of approaches, from the most basic test-to-production configuration to complex multiple test and production library configurations on multiple systems.

***TURNOVER® for iSeries Supplement #63*: *Managing Multiple Versions*** discusses the more advanced object management techniques associated with maintaining multiple versions of a software application.

## ABOUT LIBRARY STRUCTURE

When you begin modeling your software change environments using TURNOVER® for iSeries application definitions, we suggest that, if possible, you set up your application definitions with separate libraries for source files, program objects and database files. This approach is reflected by the text and examples in this document. We recommend this approach because it offers the following benefits:

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

Generally, you will find this approach flexible and easy to manage.

# ESTABLISHING SOURCE ARCHIVE POLICIES

This topic describes some things to think about when you are deciding whether or not, and how to, archive the source code for your software applications. Some sample scenarios are also provided. (Remember: these concepts describe only source archiving; TURNOVER® for iSeries archives objects differently.)

## Points to ponder

As you consider how to set up your company's source code archiving system, you might keep these things in mind:

1. Keep it as simple as possible. The less manual intervention that is required, the better!

2. Do you have any reason to archive source at more than one application level?

   *Perhaps you would like to maintain both QA and Production source archives.*

3. How long should you retain archived source? This might require considering company policy, as well as development requirements, system resources, and other factors.

   *If you are keeping source archives at both the QA and production levels of your application, for example, you would purge your QA archive much more aggressively than your production archive.*

4. How will you structure your files?

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

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

## Some archiving examples

Examples of some archive scenarios you can consider might be as follows:

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

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

## CHAPTER 3: SETTING UP TURNOVER® FOR iSERIES

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

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

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

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

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

### Other sources of information about TURNOVER® for iSeries archiving

Here are some other places in TURNOVER® for iSeries documentation where you can find archiving information:

- Learn about retrieving archive information, and read a general description of how archiving works, in ***Chapter 4*** of the ***TURNOVER® for iSeries User Guide***.
- Read about the ***Work with Source Archive Files*** feature in ***Chapter 8*** of the ***TURNOVER® for iSeries User Guide***.
- TURNOVER® for iSeries Administrator online help describes archiving utility features.
- On the greenscreen, **F1** will present field-specific help on any archiving-related field.

## CHOOSING YOUR TURNOVER® FOR iSERIES ADMINISTRATOR

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

The real objective of a TURNOVER® for iSeries system is to ensure that your department operates efficiently and that your production environment is stable and secure. TURNOVER® for iSeries does not ensure that you have adequately tested your programs. If used properly, it will improve productivity and ensure that your technical staff cannot change objects casually and, when changes are made, a permanent audit trail exists. TURNOVER® for iSeries is an enabler – it is not a substitute for good practices or qualified technical and administrative personnel. TURNOVER® for iSeries will help your administrator to be more effective, but it is not a substitute for an effective administrator.

Your TURNOVER® for iSeries Administrator could be a Database Administrator, Lead Programmer, Operations Manager, Quality Assurance Specialist, or the Information Systems Manager. In the course of his or her job, that person may perform any of the following tasks:

- Enter Change request and Problem Reports
- Evaluate Change request and Problem Reports
- Set up TURNOVER® for iSeries on your system
- Enter information about applications
- Maintain library and object authority
- Maintain TURNOVER® for iSeries user authority
- Maintain Programmer Information
- Verify plans for changes
- Review and certify test results

## CHAPTER 3: SETTING UP TURNOVER® FOR iSERIES

- Enter TURNOVER® for iSeries forms (a promotion request)
- Review TURNOVER® for iSeries forms entered by developers
- Run promotions and check the results
- Distribute promotions to remote systems
- Authorize emergency program fixes
- Produce reports and monitor change history
- Audit change history on your system.

These tasks may be shared by, or distributed among, several people. Taken together, these tasks make up what is referred to in this guide as *TURNOVER® for iSeries Administration*. You may choose not to do everything listed.

## WHERE DO TURNOVER® FOR iSERIES PROJECTS COME IN?

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

## DESIGNING YOUR TURNOVER® FOR iSERIES CHANGE ENVIRONMENTS

In this topic, we illustrate several common scenarios that represent the wide range of choices you have available. Remember, you can define virtually any configuration to TURNOVER® for iSeries.[5]

Do you have individual test libraries, or shared test libraries? Do you store production source code in object libraries? Do you store program objects in the same library as data objects? Does the answer differ by application?

### Example 1: A Simple Change Environment

Suppose you run a software application that is not mission critical and that changes only occasionally. For software like this, one application level may be appropriate. See Figure 3.

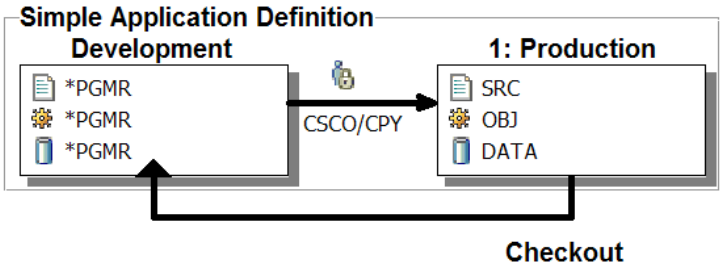



**Figure 3: A Simple Change Environment**

[5] If you do not see a solution that matches your requirements, call a UNICOM Systems, Inc. Technical Support Representative to discuss your configuration needs.

## Example 2: A More Complex Change Environment

Perhaps you use another software application that *is* mission critical. Maybe changes to this application are distributed to hundreds of users all over your company. For a more sensitive software application, three or four application definition levels might be more appropriate. See Figure 4.

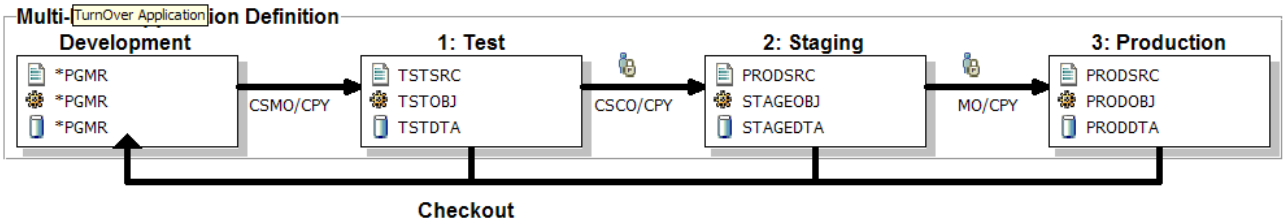



**Figure 4: A More Complex Change Environment**

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

## Example 3: An Environment for Vendor Software

Suppose another of your software applications is supplied by a software vendor, and your company makes custom modifications to the vendor base. In this scenario, you need to ensure that your changes remain separate from the vendor's base. In your TURNOVER® for iSeries application definition, you would define the application level containing the vendor's base code as ***Permanently Locked***. See Figure 5.

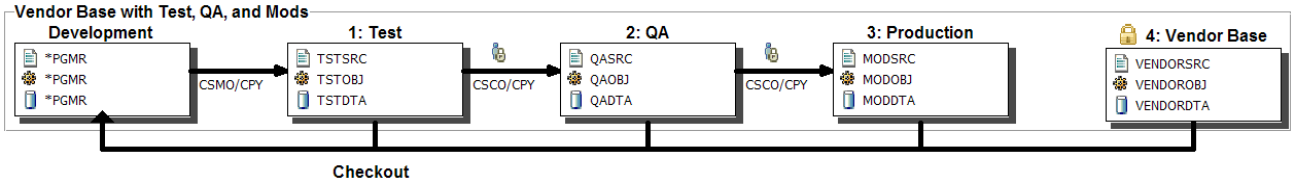


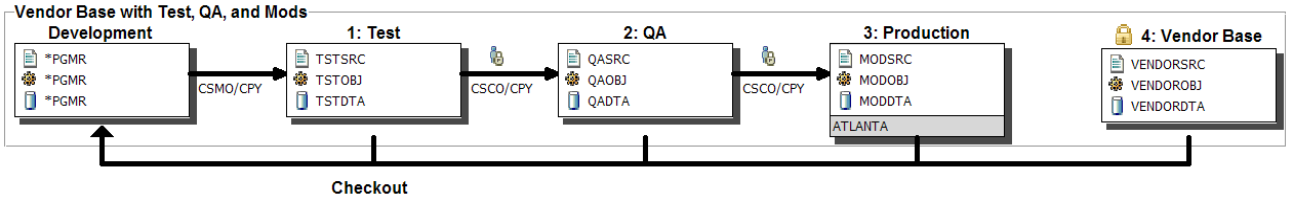

**Figure 5: A Change Environment with Vendor Base Code**

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

## Example 4: Distributing Changes to your Vendor Software

If you distribute changes to one or more computers or partitions,[6] you can do so easily when you have a remote version of TURNOVER® for iSeries running on that system. Generally, you distribute changes from the production level after objects are promoted there; however, you can distribute from any level if you choose.

Suppose you add distribution to Level 2 of Figure 5 above. Your diagram now looks like this:




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

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

[6] For more information, see ***Chapter 11: Distributing Changes to Production Computers*** in the ***TURNOVER® for iSeries User Guide*** and ***TURNOVER® for iSeries Supplement #31: Distributing to Logical Partitions***.

For the application illustrated in Figure 5, we would need a remote application definition like this on the ATLANTA system:

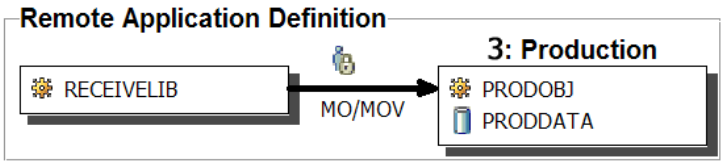



**Figure 6: Distributing Changes with a Remote Application Definition**

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

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

If you need to manage several releases and/or maintain customized version of software for several locations or customers, read ***TURNOVER® for iSeries Supplement #63: Managing Multiple Versions***.

## MODELING MADE EASY: TURNOVER® FOR iSERIES' APPLICATION DEFINITION MODELS

Your TURNOVER® for iSeries software comes complete with a set of pre-configured application definition models (not all of which were demonstrated above) that you can copy and customize to accelerate your TURNOVER® for iSeries setup phase.

If you model your applications after one of these proven models we provide, you will be up and running quickly. Even if one of our models isn't a perfect fit, adapting one to your needs is also easy to do.[7]

[7] UNICOM Systems, Inc. can also provide additional models to meet your special needs. Call our Technical Support line to discuss your application's requirements.

### UNICOM Systems, Inc. Recommends

While you can change our models and copy them to build your application definitions, we do recommend that you **preserve our models as they were shipped**, so that you always have a complete set of original models to copy and customize.

In addition, you can create application definitions "from scratch" using the TURNOVER® for iSeries Client, if you prefer. The wizards guide you through specifying the information needed to create an application definition. Generally, though, copying a model that has been customized to reflect your company's preferences and requirements is the most efficient way to proceed, because less re-configuration will be required.

By far the easiest way to proceed is to examine the models we provide, which are described in the topics that follow, and then copy and customize them to reflect the decisions you've made about how things should work at your company.

## Please read this section carefully before you start.

## CHAPTER 3: SETTING UP TURNOVER® FOR iSERIES

**To define your application**, do the following:

1. Review the schematics and pick one that fits best.
2. Diagram your own setup on paper, if you like.
3. If you intend to use TURNOVER® for iSeries' project system, then use the TURNOVER® for iSeries Client to create a project, and add tasks.
4. Test your application definitions thoroughly.

**Before you begin**, we suggest that you do the following:

1. If you plan to distribute changes, install TURNOVER® for iSeries Remote on the other computers in your network, and **define those systems on your development computer**. (To define systems in the TURNOVER® for iSeries Client, expand the Administration section, right-click Distribution Systems, and choose **New System** on the context menu.[8]) Later, when you have created your remote application definitions, you can use the Remote Configuration Wizard to send your system and remote application definitions to your remote computer(s).
2. Copy and customize the model application definitions to reflect your setup requirements. (In the TURNOVER® for iSeries Client, expand the Applications section, right-click an application model and choose **Copy application**.)
3. If your software applications require a programming language other than RPG or object types that are not included in the models, update the type code list in your customized model as necessary. (In edit mode in the Application Designer, right-click an application level and choose **Add type codes**.)
4. Enroll TURNOVER® for iSeries users (In the TURNOVER® for iSeries Client, expand the Administration section, right click on Users and choose **New User** from the context sensitive menu).

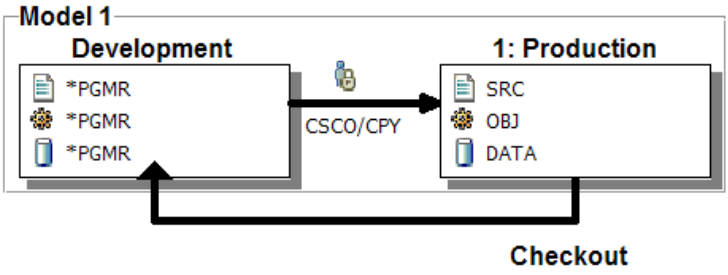
TURNOVER® for iSeries' model application definitions are set for a typical RPG development environment and assume a single computer. Performing these steps will save time later.

What do you do if none of the models fits? Pick the one that is closest, then copy and modify it to meet your needs. Or, if none of them come close, call us. We will be glad to help you define your applications.

Can you use your own application definition as a model? Yes. If you have already created an application definition that you want to use as a model, this procedure will work for you.

[8] In the TURNOVER® for iSeries Client, you will need a TURNOVER® for iSeries connection for each of your remote systems to create system definitions on those computers.

## Model 1 (01M)




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

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

   New object names are reserved in TURNOVER® for iSeries.

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

2. Programmer makes changes to source, and compiles and tests it in his/her library.

3. Programmer builds TURNOVER® for iSeries form and runs error-check.

4. Form is approved.

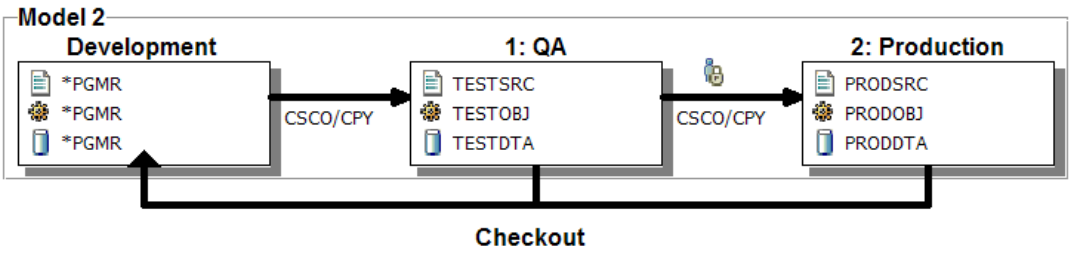
5. Form is run. Source is copied into SRC library and compiled into OBJ library.
   **Note:** Library list for level 1 form is:
   SRC
   OBJ
   DATA
   :
   :

   IFS objects are moved or copied.

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

   The programmer's library contains only changed objects. They are deleted after the TURNOVER® for iSeries form runs.

## Model 2 (02M)




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

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

   New object names are reserved in TURNOVER® for iSeries.

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

2. Programmer makes changes to source, and compiles and tests it in his/her library.

3. Programmer builds TURNOVER® for iSeries level 1 form and runs error-check.

4. Level 1 form is approved.

5. Level 1 form is run. Source is copied into TESTSRC library and compiled into TESTOBJ. IFS objects are moved or copied.

   **Note:** Library list for level 1 form is:
   TESTSRC
   TESTOBJ
   TESTDTA
   PRODSRC
   PRODOBJ
   PRODDTA
   :
   :
   Programmer's development library should not be included in the library list.

   TESTOBJ and TESTDTA contain only changed objects.

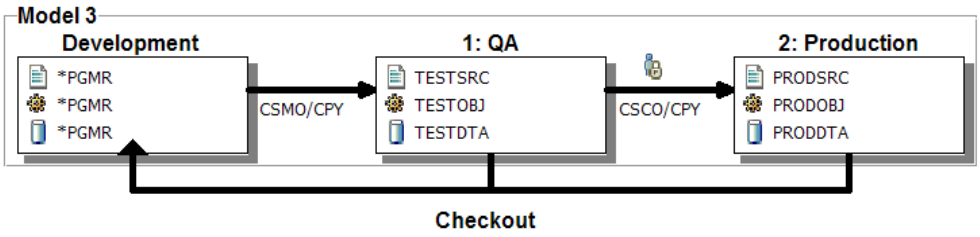
6. TURNOVER® for iSeries administrator copies one or more level 1 forms to level 2 (preparing them to run for level 2).

7. Level 2 form is approved and run. TURNOVER® for iSeries copies source into PRODSRC and compiles objects into PRODOBJ and PRODDTA. LFs are recreated over PFs. Old source is archived. Old object is archived (optional). Source and object are recorded as having been checked in. IFS objects are moved or copied.

   **Note:** Library list for level 2 run is:
   PRODSRC
   PRODOBJ
   PRODDTA
   :
   :
   This library list is all that should be needed to create objects unless some programs refer to objects in other libraries.

## Model 3 (03M)




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

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

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

2. Programmer makes changes to source and compiles and tests it in his/her library.

3. Programmer builds TURNOVER® for iSeries level 1 form and runs error-check.

4. Level 1 form is approved.

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

   **Note:** Library list for level 1 form is:
   TESTSRC
   TESTOBJ
   TESTDTA
   PRODOBJ
   :
   Programmer's development library should not be included in the library list.

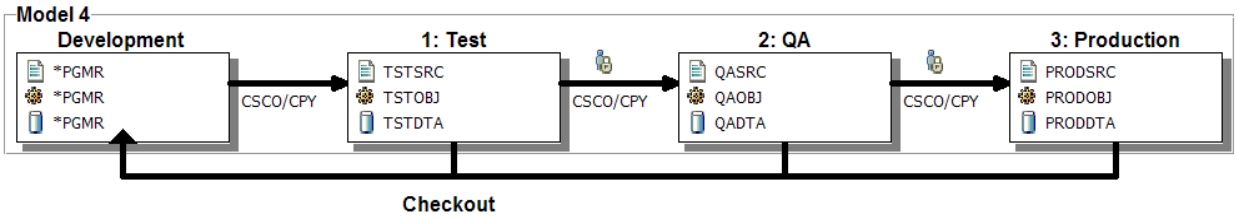
   TESTOBJ and TESTDTA contain only changed objects.

6. TURNOVER® for iSeries administrator copies one or more level 1 forms to level 2 (preparing them to run for level 2).

7. Level 2 form is approved and run. TURNOVER® for iSeries copies source into PRODSRC and compiles objects into PRODOBJ and PRODDTA. IFS objects are moved or copied. LFs are recreated over PFs. Existing data is copied to new files. Old source is archived. Old object is archived (optional). Source is recorded as having been checked in.

   **Note:** Library list for level 2 run is:
   PRODSRC
   PRODOBJ
   PRODDTA
   :
   :
   This library list is all that should be needed to create objects unless some programs refer to objects in other libraries.

## Model 4 (04M)




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

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

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

2. Programmer makes changes to source and compiles and tests it in his/her library.

3. Programmer builds TURNOVER® for iSeries level 1 form and runs error-check.

4. Level 1 form is approved and run. TURNOVER® for iSeries copies source into TSTSRC library and compiles objects into TSTOBJ and TSTDTA. IFS objects are moved or copied.

   **Note:** Library list for level 1 form is:

   | | |
   |---|---|
   | TSTSRC | QADTA |
   | TSTOBJ | PRODSRC |
   | TSTDTA | PRODOBJ |
   | PRODDTA | |
   | QASRC | : |
   | QAOBJ | : |

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

   TSTOBJ and TSTDTA contain only changed objects.

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

6. Level 2 form is approved.

7. Level 2 form is run. TURNOVER® for iSeries copies source into QASRC and compiles objects into QAOBJ and QADTA. IFS objects are moved or copied.

   **Note:** Library list for level 2 run is:

   QASRC
   QAOBJ
   QADTA
   PRODSRC
   PRODOBJ
   PRODDTA

8. TURNOVER® for iSeries Administrator copies one or more Level 2 forms to Level 3 (preparing them to run for Level 3).

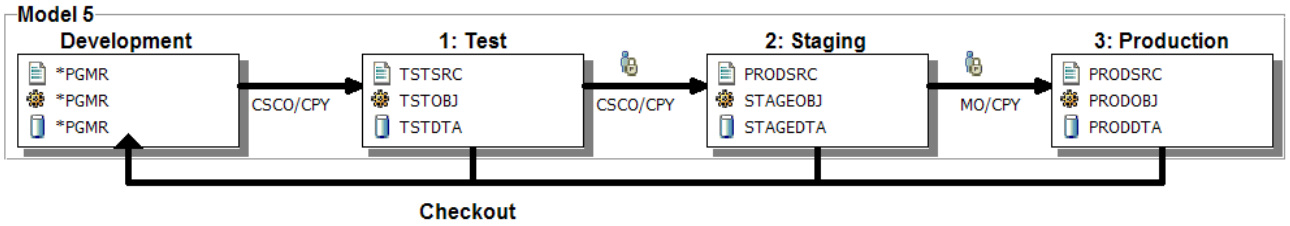
9. Level 3 form is approved and run. TURNOVER® for iSeries copies source into PRODSRC and compiles objects into PRODOBJ and PRODDTA. IFS objects are moved or copied. LFs are recreated over PFs. Old source is archived. Old object is archived (optional). Source is recorded as having been checked in.

   **Note:** Library list for Level 3 run is:

   PRODSRC
   PRODOBJ
   PRODDTA
   :
   :

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

## Model 5 (05M)




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

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

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

2. Programmer makes changes to source and compiles and tests it in his/her library.

3. Programmer builds TURNOVER® for iSeries level 1 form and runs error-check.

4. Level 1 form is approved and run. TURNOVER® for iSeries copies source into TSTSRC library and compiles objects into TSTOBJ and TSTDTA. IFS objects are moved or copied.

   **Note:** Library list for level 1 form is:

   | | |
   |---|---|
   | TSTSRC | PRODSRC |
   | TSTOBJ | PRODOBJ |
   | TSTDTA | PRODDTA |
   | : | |
   | STAGEOBJ | : |
   | STAGEDTA | |

   Programmer's development library should <u>not</u> be included in the library list.

   TSTOBJ and TSTDTA contain only <u>changed</u> objects.

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

6. Level 2 form is approved.

7. Level 2 form is run. Copy source into PRODSRC and compile objects into STAGEOBJ and STAGEDTA. IFS objects are moved or copied.
   **Note:** Library list for level 2 run is:
   PRODSRC
   STAGEOBJ
   STAGEDTA
   PRODOBJ
   PRODDTA

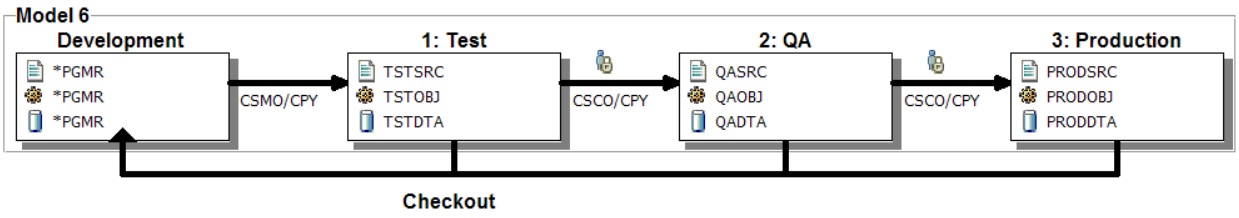
8. TURNOVER® for iSeries Administrator copies one or more Level 2 forms to Level 3 (preparing them to run for Level 3).

9. Level 3 form is approved and run. Objects are moved into PRODOBJ and PRODDTA. IFS objects are moved or copied. LFs are recreated over PFs. Old source is archived. Old object is archived (optional). Source is recorded as having been checked in. Data is copied to new files.

   **Note:** Library list for Level 3 run is:
   PRODSRC
   PRODOBJ
   PRODDTA
   :
   :

   This library list is all that should be needed to create objects unless some programs refer to objects in other libraries.
   **Yes – the source library at level 2 and 3 are the same name. This is intentional.

## Model 6 (06M)




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

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

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

2. Programmer makes changes to source and compiles and tests it in his/her library.

3. Programmer builds TURNOVER® for iSeries level 1 form and runs error-check.

4. Level 1 form is approved and run. TURNOVER® for iSeries copies source into TSTSRC library and moves objects into TSTOBJ and TSTDTA. IFS objects are moved or copied.

   **Note:** Library list for level 1 form is:

   | | |
   |---|---|
   | TSTSRC | PRODSRC |
   | TSTOBJ | PRODOBJ |
   | TSTDTA | PRODDTA |
   | QASRC | |
   | QAOBJ | : |
   | QADTA | : |

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

   TSTOBJ and TSTDTA contain only changed objects.

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

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

   **Note:** Library list for level 2 run is:

   QASRC
   QAOBJ
   QADTA
   PRODSRC
   PRODOBJ
   PRODDTA
   :
   :

7. TURNOVER® for iSeries Administrator copies one or more Level 2 forms to Level 3 (preparing them to run for Level 3).

8. Level 3 form is approved and run. TURNOVER® for iSeries copies source into PRODSRC and compiles objects into PRODOBJ and PRODDTA. IFS objects are moved or copied. LFs are recreated over PFs. Old source is archived. Old object is archived (optional). Source is recorded as having been checked in.

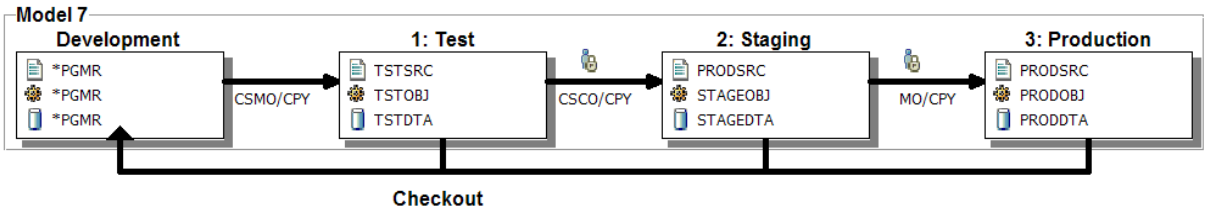
   **Note:** Library list for Level 3 run is:

   PRODSRC
   PRODOBJ
   PRODDTA
   :
   :

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

## Model 7 (07M)




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

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

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

2. Programmer makes changes to source and compiles and tests it in his/her library.

3. Programmer builds TURNOVER® for iSeries level 1 form and runs error-check.

4. Level 1 form is approved and run. TURNOVER® for iSeries copies source into TESTSRC library and moves objects into TESTOBJ. IFS objects are moved or copied.

   **Note:** Library list for level 1 form is:
   TESTSRC
   TESTOBJ
   TESTDTA
   :
   :

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

   TESTOBJ and TESTDTA contain only changed objects.

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

6. Level 2 form is approved and run. TURNOVER® for iSeries copies source into PRODSRC and compiles objects into STAGEOBJ and STAGEDTA. IFS objects are moved or copied. Old source is archived.

   **Note:** Library list for level 2 run is:
   PRODSRC
   STAGEOBJ
   STAGEDTA
   PRODSRC
   PRODOBJ
   PRODDTA
   :
   :

7. TURNOVER® for iSeries Administrator copies one or more Level 2 forms to Level 3 (preparing them to run for Level 3).

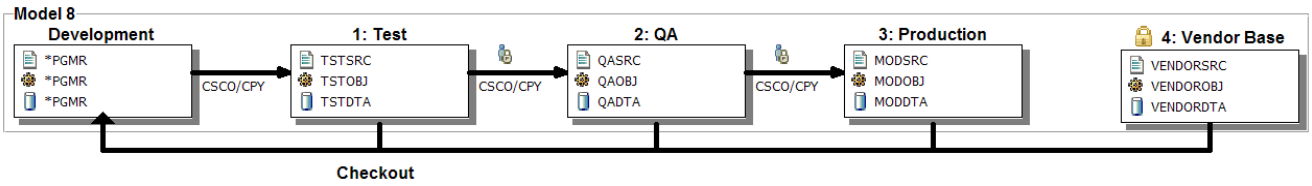
10. Level 3 form is approved and run. TURNOVER® for iSeries moves objects into PRODOBJ and PRODDTA. IFS objects are moved or copied. LFs are recreated over PFs. Old object is archived (optional). Source is recorded as having been checked in. Existing data is copied to new files.

    **Note:** Library list for Level 3 run is:
    PRODSRC
    PRODOBJ
    PRODDTA
    :
    :

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

## CHAPTER 3: SETTING UP TURNOVER® FOR iSERIES

# Model 8 (08M): VENDOR PACKAGE MODEL A




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

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

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

2. Programmer makes changes to source and compiles and tests it in his/her library.

3. Programmer builds TURNOVER® for iSeries level 1 form and runs error-check.

4. Level 1 form is approved and run. TURNOVER® for iSeries copies source into TESTSRC library and compiles objects into TESTOBJ and TESTDTA. IFS objects are moved or copied.

   **Note:** Library list for level 1 form is:

   | | |
   |---|---|
   | TESTSRC | MODSRC |
   | TESTOBJ | MODOBJ |
   | TESTDTA | MODDTA |
   | QASRC | VENDORSRC |
   | QAOBJ | VENDOROBJ |
   | QADTA | VENDORDTA |

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

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

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

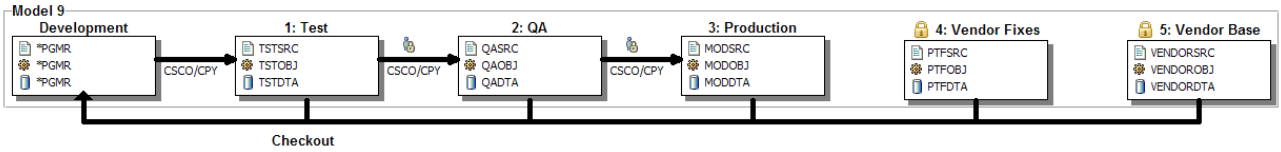
   **Note:** Library list for level 2 run is:
   QASRC
   QAOBJ
   QADTA
   MODSRC
   MODOBJ
   MODDTA
   VENDORSRC
   VENDOROBJ
   VENDORDTA

7. TURNOVER® for iSeries Administrator copies one or more Level 2 forms to Level 3 (preparing them to run for Level 3).

8. Level 3 form is approved and run. TURNOVER® for iSeries copies source into MODSRC and compiles objects into MODOBJ and MODDTA. IFS objects are moved or copied. LFs are recreated over PFs. Old source is archived. Old object is archived (optional). Source is recorded as having been checked in, even if it had been copied from VENDORSRC.

   **Note:** Library list for Level 3 run is:
   MODSRC
   MODOBJ
   MODDTA
   VENDORSRC
   VENDOROBJ
   VENDORDTA
   This library list is all that should be needed to create objects unless some programs refer to objects in other libraries.

## Model 9 (09M): VENDOR PACKAGE MODEL B




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

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

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

2. Programmer makes changes to source and compiles and tests it in his/her library.

3. Programmer builds TURNOVER® for iSeries level 1 form and runs error-check.

4. Level 1 form is approved and run. TURNOVER® for iSeries copies source into TESTSRC library and compiles objects into TESTOBJ and TESTDTA. IFS objects are moved or copied.

   **Note:** Library list for level 1 form is:

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

   Programmer's development library should <u>not</u> be included in the library list.

   TESTOBJ and TESTDTA contain only objects being changed.

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

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

   **Note:** Library list for level 2 run is:

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

7. TURNOVER® for iSeries Administrator copies one or more Level 2 forms to Level 3 (preparing them to run for Level 3).

8. Level 3 form is approved and run. TURNOVER® for iSeries copies source into MODSRC and compiles objects into MODOBJ and MODDTA. IFS objects are moved or copied. LFs are recreated over PFs. Old source is archived. Old object is archived (optional). Source is recorded as having been checked in, even if had been copied from PTFSRC or VENDORSRC.

   **Note:** Library list for Level 3 run is:

   | | |
   |---|---|
   | MODSRC | PTFDTA |
   | MODOBJ | VENDORSRC |
   | MODDTA | VENDOROBJ |
   | PTFSRC | VENDORDTA |
   | PTFOBJ | : |

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

# CHAPTER 4: CREATING YOUR APPLICATION DEFINITIONS

## WHERE DO I BEGIN?

Some people define only a few applications to manage all of their objects. Some define several, each represented by a different set of libraries. Generally, you should define one application for each of your production program libraries. If all of your program objects are in one or just a few libraries, you can decide to define multiple applications anyway. You can define your TURNOVER® for iSeries applications to share the same production (target) libraries. Consider these questions when deciding how to define your applications to TURNOVER® for iSeries:

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

**UNICOM Systems, Inc. Recommends**

Start off by implementing one application definition and testing it fully. Then go on to implement others. Before you begin, make sure you've read Chapter 3 in this guide.

The procedures that follow are based on the TURNOVER® for iSeries Client Administration module.

If you have not already installed and configured that software, please do so now so you can follow along.

Refer to your ***Getting Started with TURNOVER® for iSeries Client*** guide for instructions.

## CHAPTER 4: CREATING YOUR APPLICATION DEFINITIONS

# CREATION METHODS

There are two ways to create application definitions. You can "clone" your applications from one or more of the models we provide, or build your own application definitions. The result and objective is the same: to define your software application change environment to TURNOVER® for iSeries.

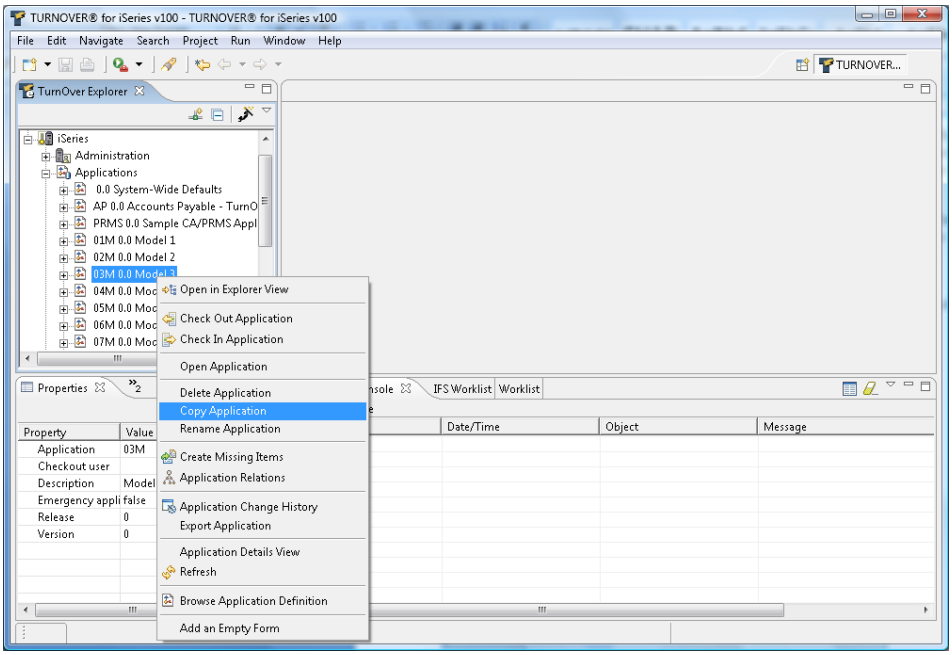
## Method 1: Copying an Application Model

In Chapter 3, you learned that you can use application models to ease the process of creating your TURNOVER® for iSeries application definitions. You learned that a model can be one of ours or one of yours. You became familiar with the models we provide, and learned the concept of copying one of our models and modifying it to create your own custom model.

At this point, you should be familiar enough with our models to be able to decide which model you would like to use. (If you haven't decided which model to use yet, but want to walk through the process anyway, then you can just choose a model to copy for your test run. You can always delete the results and start over.)

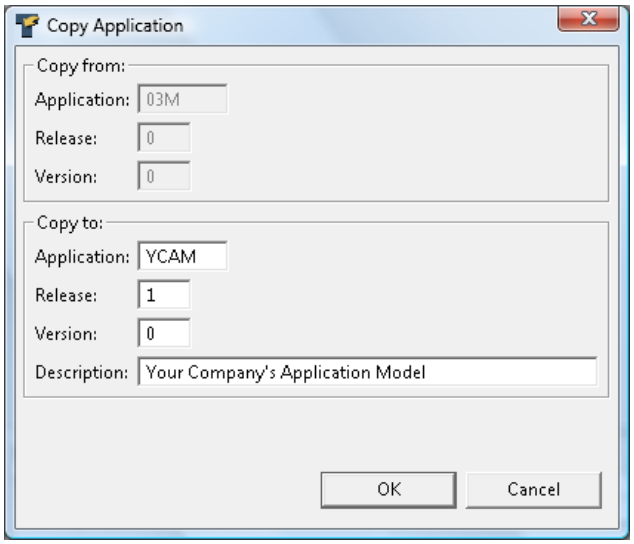
These instructions assume that you have installed and configured the TURNOVER® for iSeries Client; that you are enrolled in TURNOVER® for iSeries; and that you can see the TURNOVER® for iSeries application definition models (named 01M – 09M). We will use Model 3 in these instructions.

1. Start the TURNOVER® for iSeries Client. Under **Applications**, right-click your chosen model and choose **Copy Application**.



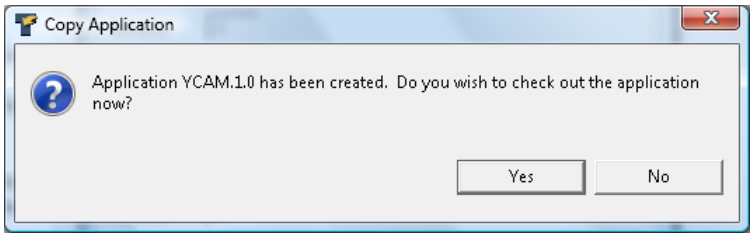

  

2. Fill out the **Copy Application** prompt, supplying your own application name, release, version, and description information.




Click **OK**.

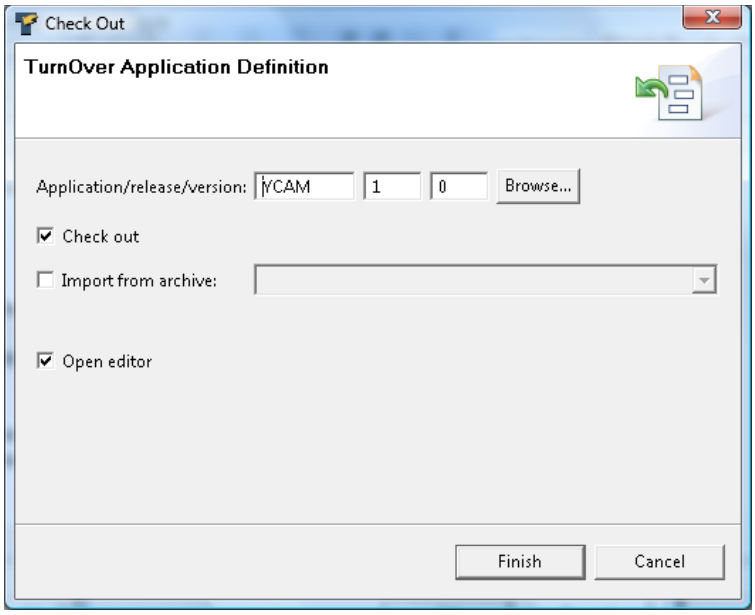
3. A progress meter appears, followed by a prompt to check out your new application definition.




Click **Yes**.

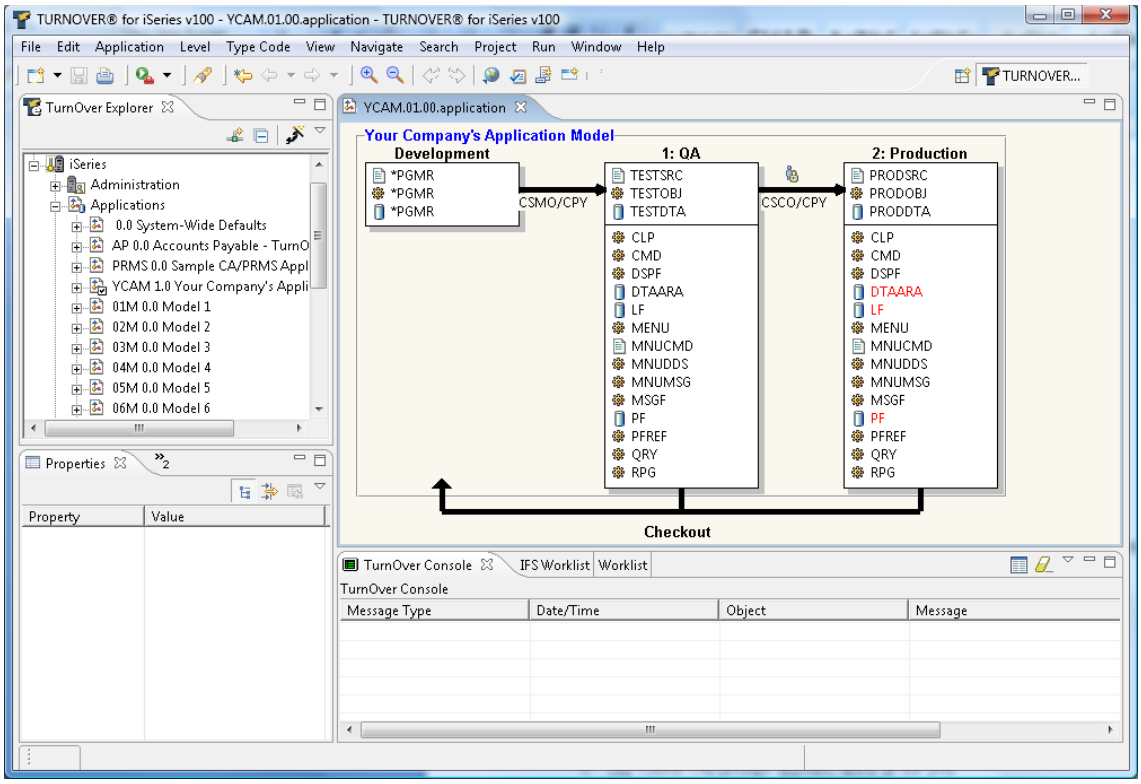
4. The Check Out prompt appears, filled in for you.

## CHAPTER 4: CREATING YOUR APPLICATION DEFINITIONS




Click **FINISH** to accept the defaults and open your new application definition in the Application Designer.

Your application definition is created, checked out, and opened in the editing pane of the TURNOVER® for iSeries Client application window. (You may need to adjust your application window size to see the entire picture.)




Congratulations! You've just created an application definition. You can now proceed with editing your custom model as necessary, to make it match your development requirements. Among the additions you might consider:

- Add explode locations to one or more type codes, to propagate changes into multiple test locations during promotions.
- Add alternate promotion paths, for objects that need to follow a slightly-different-than-default path into production.
- Add additional object type codes for other languages or platforms.
- Configure distribution at one or more levels.

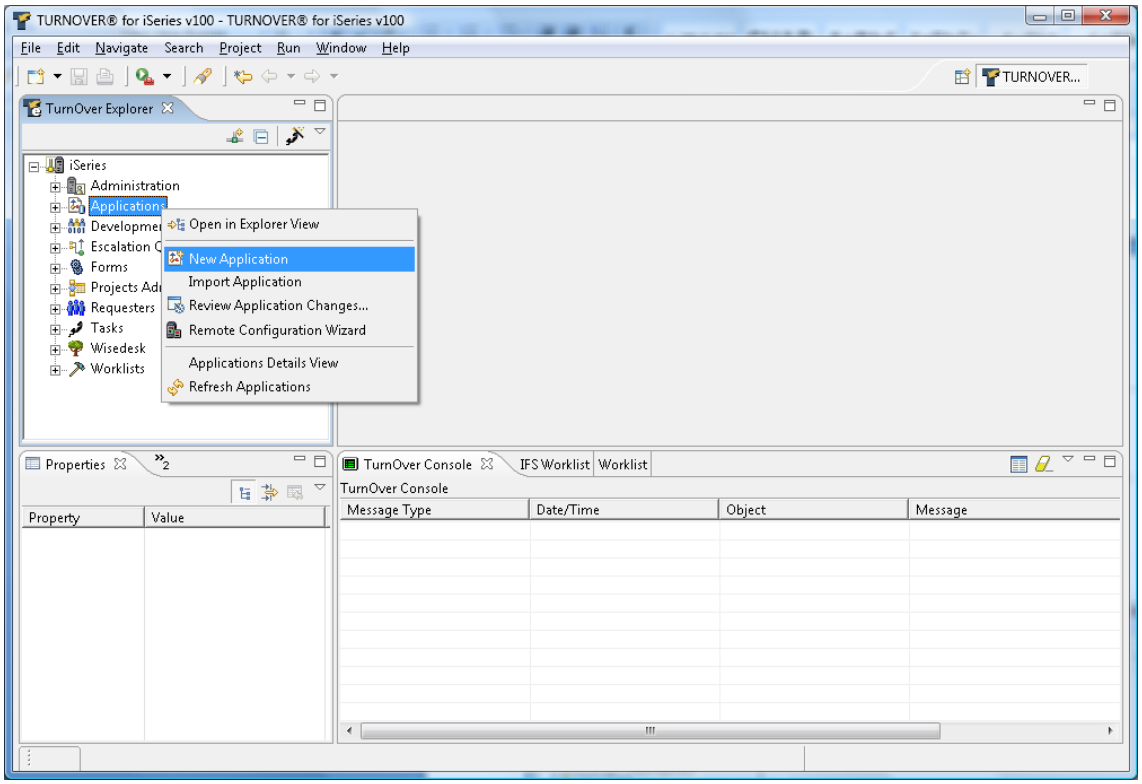
We suggest that you test your model. When you are satisfied with it, you can copy it just as you did ours, and customize it for a particular software application.

## Method 2: Using the New Application Definition wizard

With this method, the wizard guides you through the creation process. You see more of the choices that are available to you as you progress through the wizard pages.

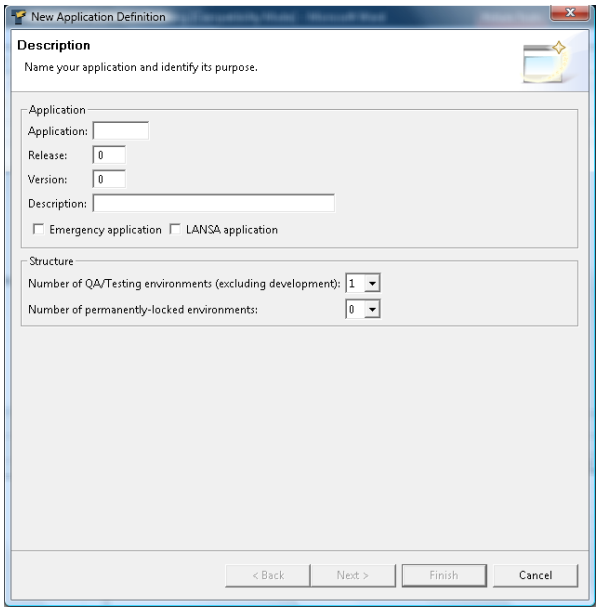
To begin, start the TURNOVER® for iSeries Client application.

1. Locate and right-click the Applications subsystem.




2. On the context menu, choose **New Application** to start the New Application Wizard.

3. On the Description page, supply information that describes the naming and structure of your planned application definition.

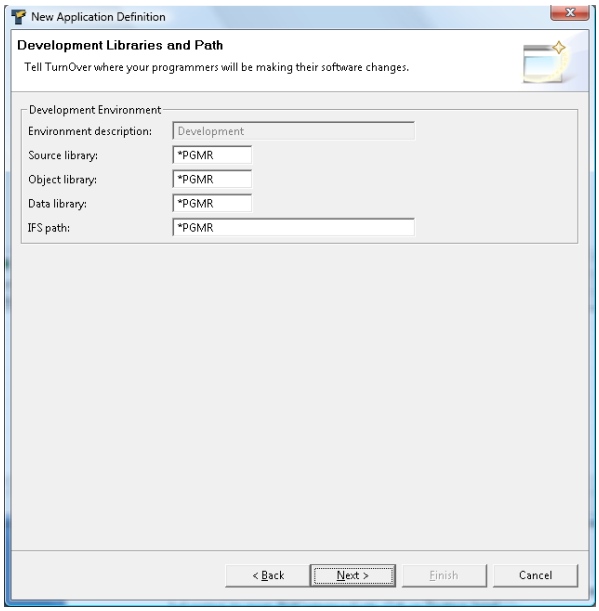



Indicate also whether this application will be used for emergency changes.

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

When you have completed the Description page to your satisfaction, click **Next** to continue.

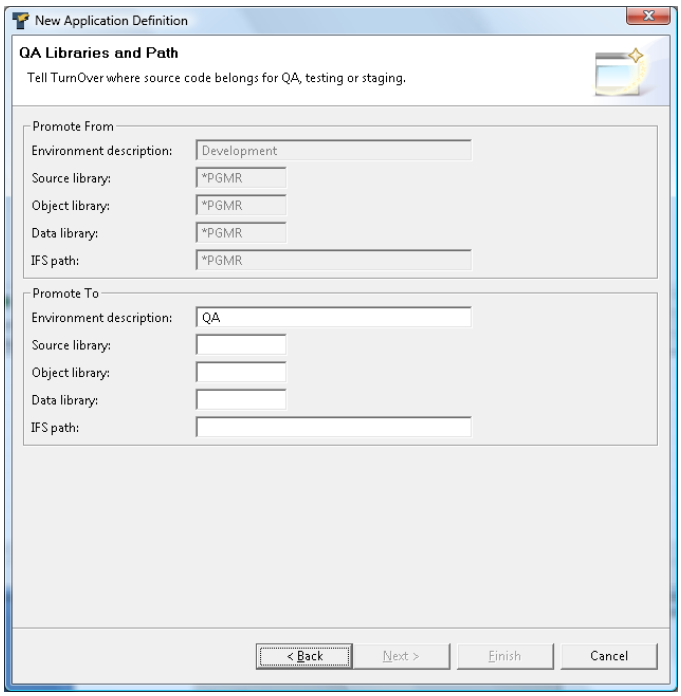
4. On the Development Libraries and Path page, identify your development source, object, and data libraries and the path for IFS objects.




The default *PGMR value tells TURNOVER® for iSeries to retrieve this information from the user's Programmer information.

Click **Next** to continue.

5. On the QA Libraries and Path page, identify the source, object, and data libraries and IFS path belonging to your first intermediate QA or Testing level.

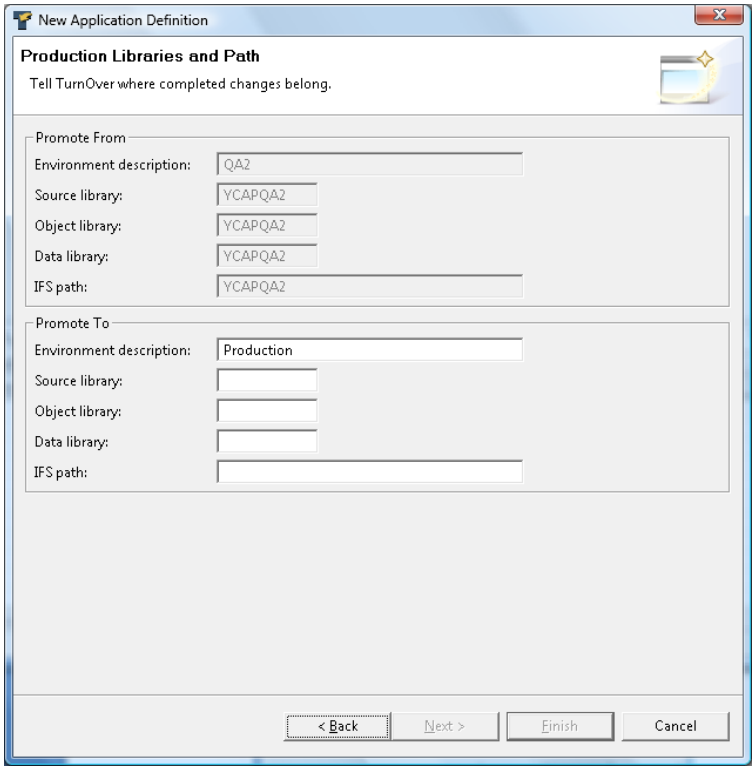


Modify the description for this level, as needed.

Click **Next** to continue.

6. Repeat the previous step for each intermediate level you specified on the Description page.

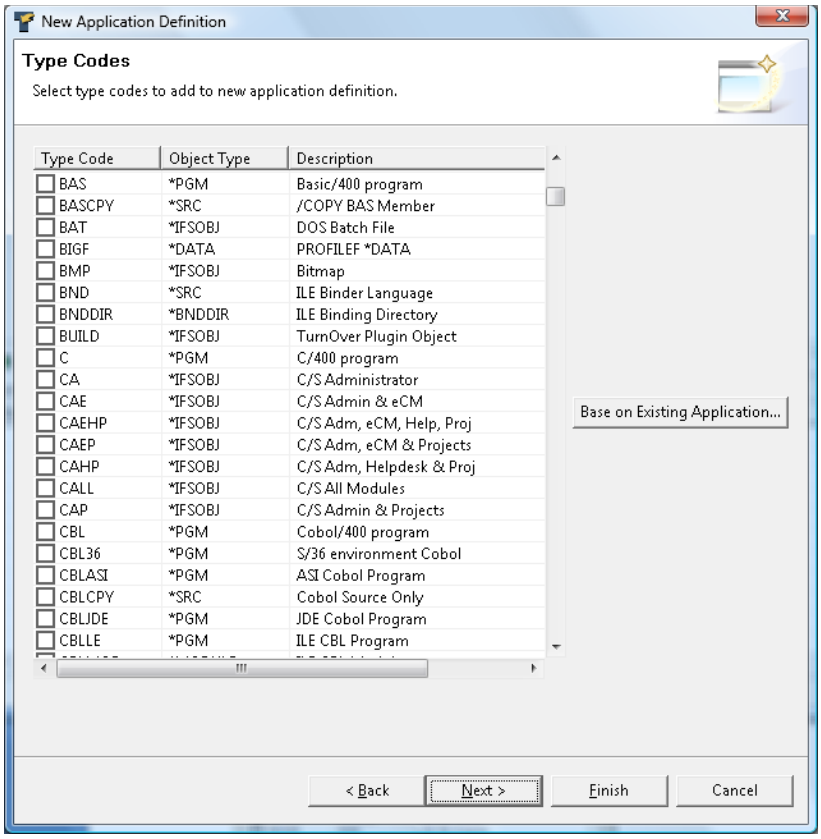
7. On the Production Libraries and Path page, identify the source, object, and data libraries and the IFS path that belong to your production environment.




Modify the description for this level, as needed.

Click **Next** to continue.

8. On the Type Codes page, identify the object types that make up your software application.

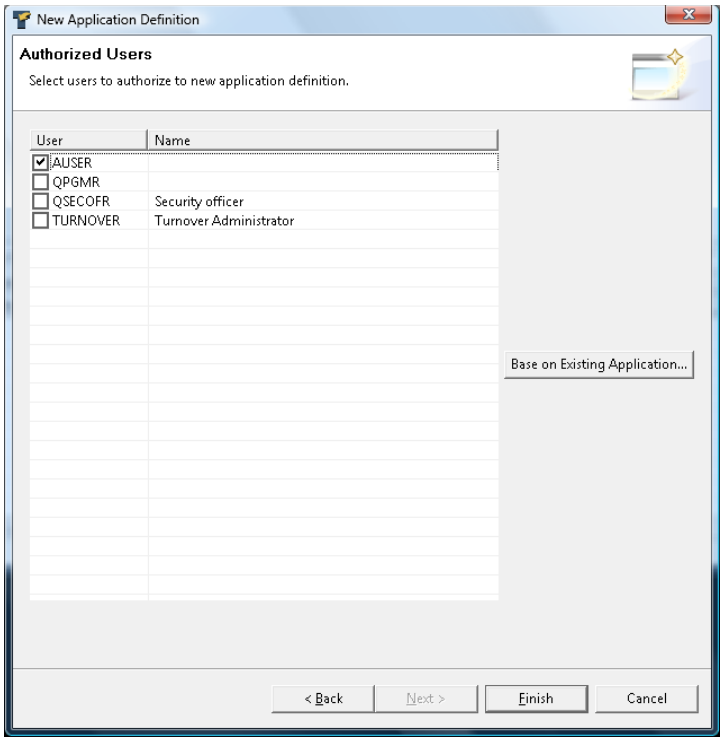



This list reflects the global type codes that are defined to your TURNOVER® for iSeries system. Check the box next to a type code to include it in all levels of your application definition.

If you find that an object type code you need does not appear in the list, make a note of this; you can create it later using Administration ▸Type Codes, and add it to your application levels after you open your application in the Application Designer.

Click **Next** to continue.

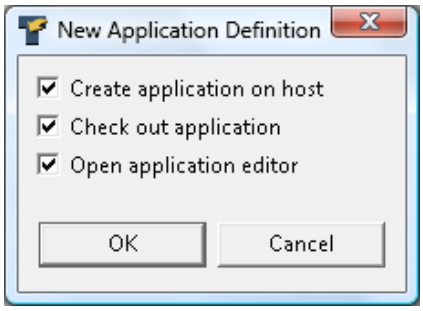
9. On the Authorized Users page, select the users who should have authority to this application definition.




If you have additional users that do not appear in this list, make a note of that now. You can enroll and authorize them later using Administration ▸ Users.

10. Click **Finish** to continue.

11. On the New Application Definition prompt, leave the defaults checked.

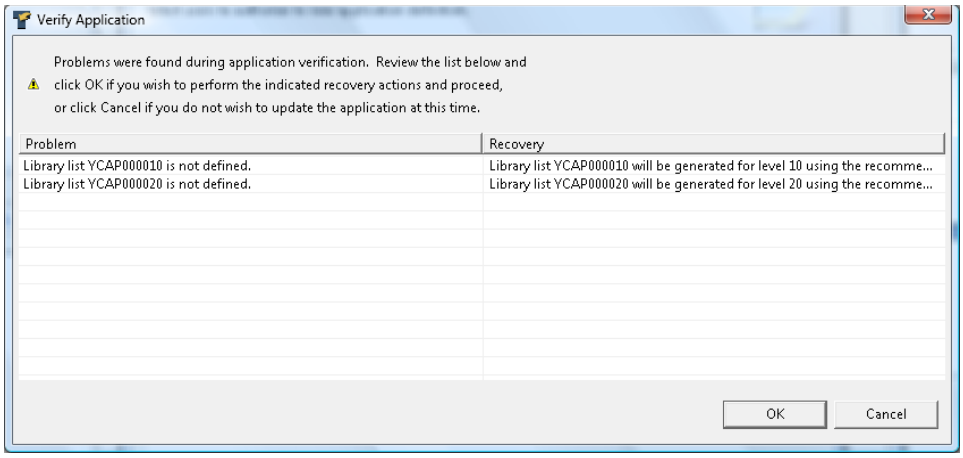



Click OK.

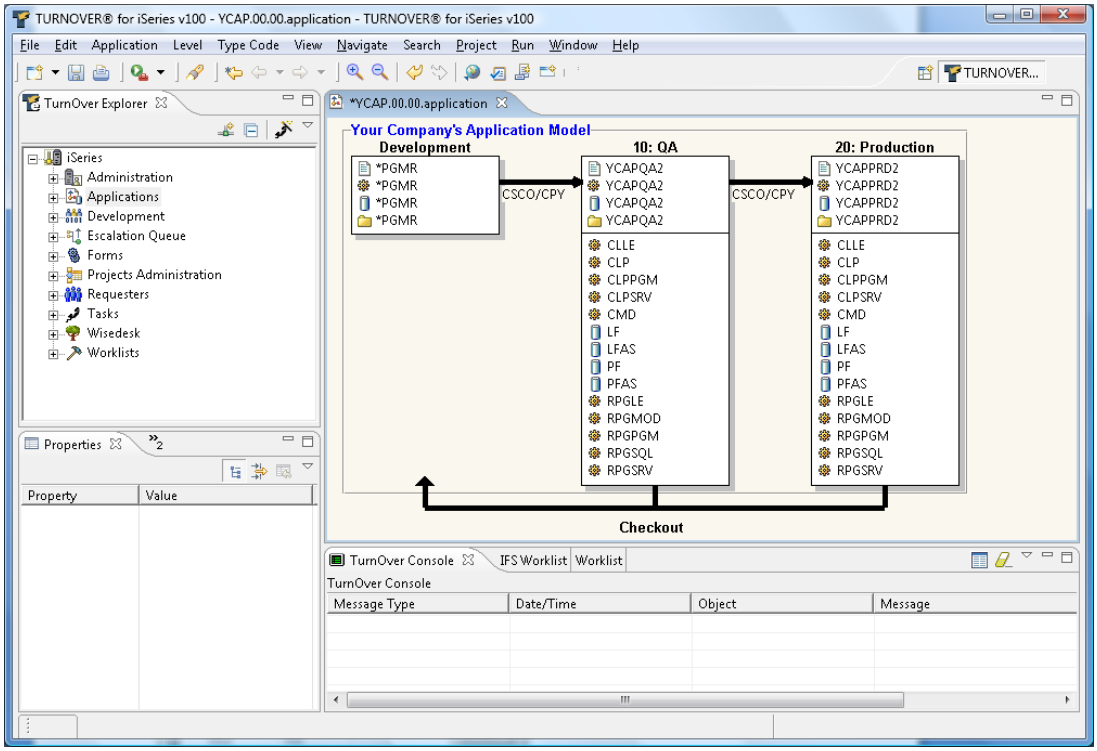
Your application definition is created on your IBM i server, checked out to you, and opened in the Application Designer.

During this process TURNOVER® for iSeries verifies your application definition. If certain necessary items do not exist, such as library list objects, TURNOVER® for iSeries will advise you and give you the option to create them at this time.

## CHAPTER 4: CREATING YOUR APPLICATION DEFINITIONS




Click **OK** to continue.




If you made a note of type codes and users you would need to add to this application definition later, you can use the Administration wizards to create those items now. After enrolling new users, you can manage their application authorities in their User Properties.

To add a type code to your application, you would return to the Application Designer, right-click a level, and choose **Add type codes** on the context menu. TURNOVER® for iSeries automatically adds the type code to all levels.

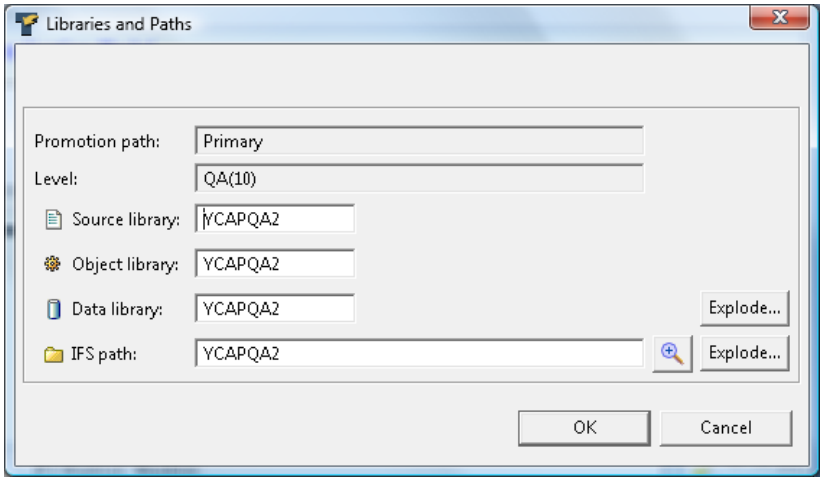
# INTRODUCTION TO THE APPLICATION DESIGNER

TURNOVER® for iSeries' new Application Designer feature lets you visualize your application definition as you create it. With Application Designer, you can tell at a glance if something is not configured properly, and correcting the problem is as easy as the click of a mouse.

In this topic, we will introduce you to some of the primary features of the Application Designer.

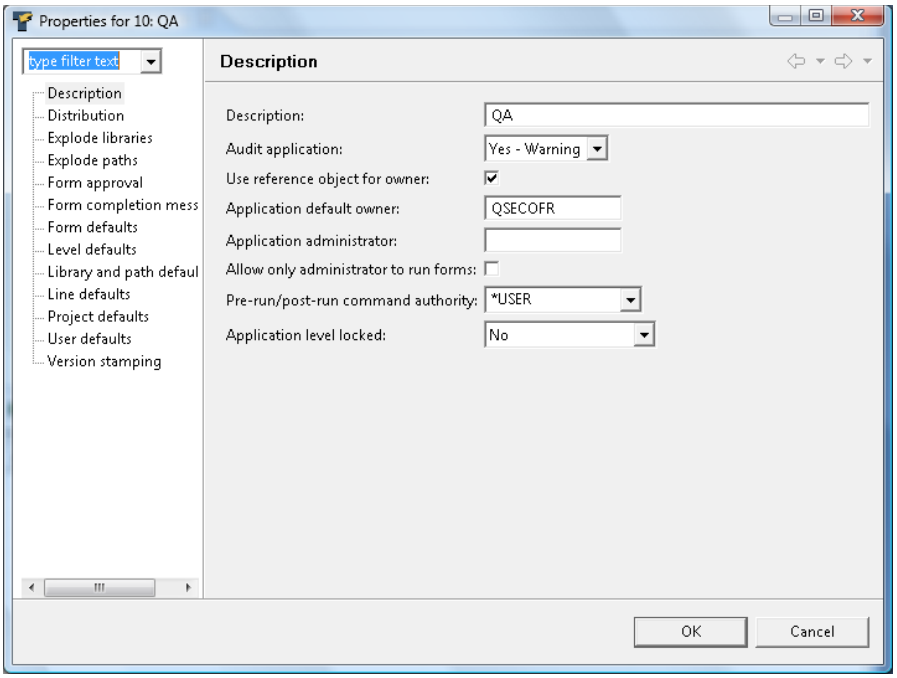
## Level Libraries and Paths

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




## Level Properties

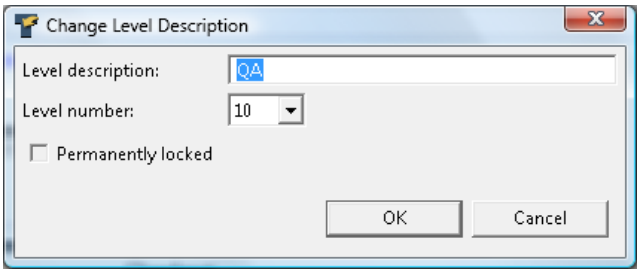
The meat of your application definition can be found in level properties. To work with the properties of an application level, right-click on or near the level in the application image and choose **Level properties** on the context menu. The resulting Level Properties dialog looks like this:




In the left pane of the panel, the list of property pages appears. When you select a page, the right pane shows the settings and values on that page.

## Level Description

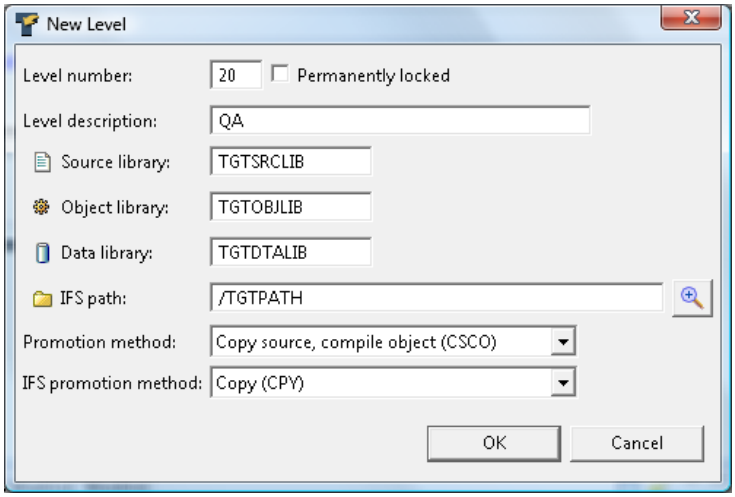
An application level's description consists of descriptive text, a level number, and a lock indicator. Access the level description by right-clicking on or near the level, and choosing **Change level description** on the context menu.




You can adjust the descriptive text at any time, but you should avoid changing the level number once the application is in use. Note that intermediate levels of an application definition cannot be permanently locked.

## Inserting Application Levels

When you create an application definition using the New Application wizard, TURNOVER® for iSeries automatically leaves space between levels, to allow you to insert new levels later with minimal disruption. To insert an application level, right-click a level. Depending on the level you chose, the context menu contains an option for inserting a level before, or inserting a level after the selected level.

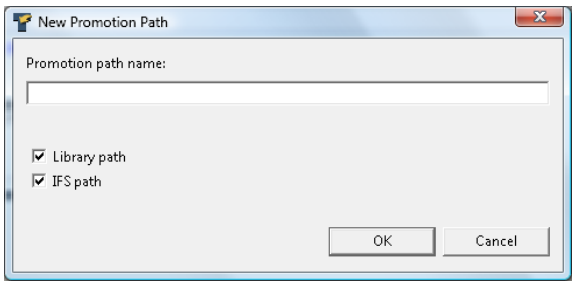



Once you fill in the New Level prompt, TURNOVER® for iSeries inserts a new application level, numbering it as you designate. TURNOVER® for iSeries warns you if inserting the designated level number will cause higher levels to be re-numbered, so you can adjust the level number you specified if necessary.

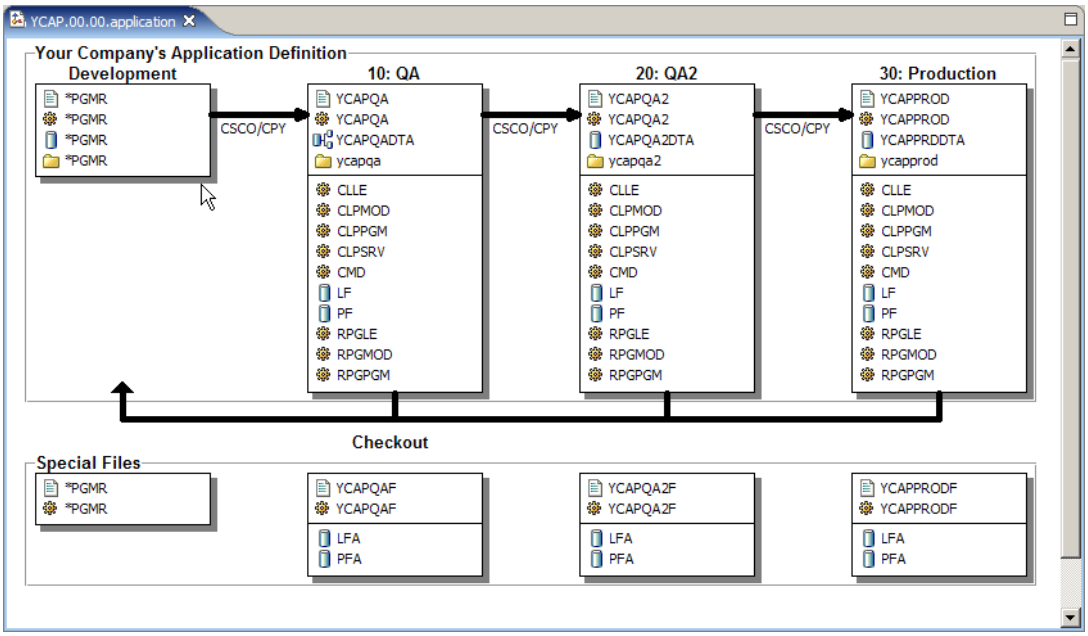
## Promotion Paths

Before Application Designer, if you needed a particular object type to follow a non-default path through the application to Production, you had to customize the type code at the application level.

With Application Designer, a new concept called Promotion Paths makes this easier to do. When you want a set of objects to be stored in a different location than other objects, you create something called a Promotion Path. To create a new promotion path, right-click anywhere in your application diagram and choose **New promotion path**.



Give your promotion path a name and indicate whether it has libraries or IFS paths (or both) associated with it. Once created, you would modify the libraries or paths associated with the promotion path, then assign type codes to it. This is how a new promotion path looks when added to your application diagram and populated with some type codes:

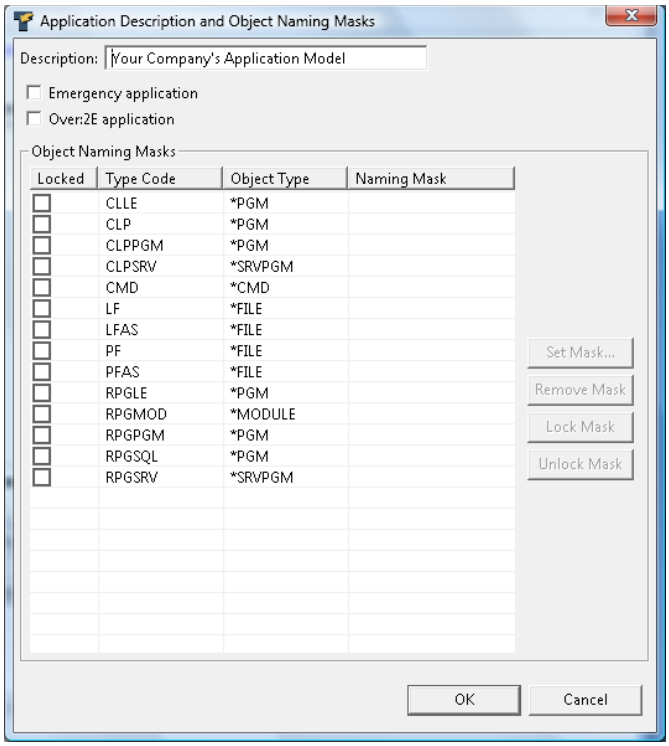



## Application Description and Object Naming Masks

If you plan to require that software object names adhere to a specific naming structure, you can define that in the Application Description. This is also where you maintain the primary text description for your application definition, as well as whether it is used for emergency changes.

To access the application description, right click anywhere in your application diagram and choose **Application description and object naming masks**.



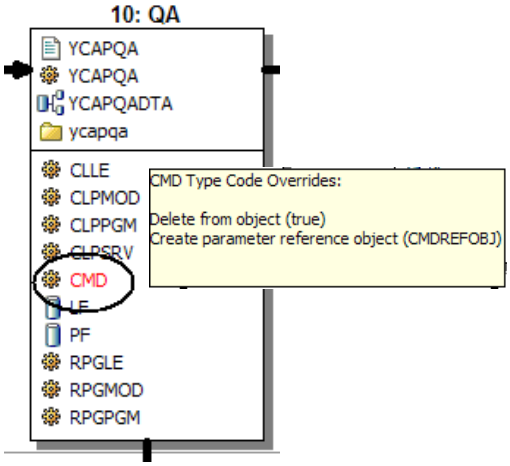

## Type Code Overrides

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

After you've made your changes, the type code name changes color in the diagram.

When you hover your cursor over a customized type code, the customized values appear in a popup.

The picture at right shows how a customized type code looks in the Application Designer.[9]




[9] To manage the colors used by the Application Designer, choose Preferences on the Window menu. Then, under the TURNOVER® for iSeries category, expand Administrator and go to the Application Designer page.

# CHAPTER 4: CREATING YOUR APPLICATION DEFINITIONS

# Appendix A: TURNOVER® for iSeries Configuration Checklists

### UNICOM Systems, Inc. Recommends

We mention both interfaces here, but we recommend that you use the TURNOVER® for iSeries Client module.

## GETTING STARTED CHECKLIST

Use this checklist to help you figure out what decisions to make. You should make these decisions first**.**

Change the company name field. (Helps you keep track of what system you are on.) ❑

**In the Client**: The company name default is on the Global Defaults – Miscellaneous page. Expand Administration, then Utilities; then choose Global Defaults – Miscellaneous in the left pane.
**On the IBM i**: Choose Main Menu option **8**, then option **5** (roll down).

Who will be assigned as the TURNOVER® for iSeries Administrator? ❑

See *CHOOSING YOUR TURNOVER® FOR iSERIES ADMINISTRATOR* on page 3-5**.**

What applications am I going to define first? ❑

Which TURNOVER® for iSeries application model meets my software application's requirements best? (Repeat this evaluation for each software application you will manage) ❑

See ***CHAPTER 3:** SETTING UP* in this guide.

What TURNOVER® for iSeries projects will I need? ❑

See ***Chapter 9: Working with Projects and Tasks*** in the ***TURNOVER® for iSeries User Guide***.

Which programmers and other users will be changing the application? ❑

Which programmers and others will have access to the project system? ❑

To which systems will I be distributing changes for this software application? ❑

## Appendix A: TURNOVER® for iSeries Configuration Checklists

# GENERAL SETUP CHECKLIST

### Then you can begin to set up TURNOVER® for iSeries:[10]

Enroll and authorize TURNOVER® for iSeries users - Main Menu option **3** ❑
See ***Chapter 3: Authorizing Users to*** *TURNOVER® for iSeries* in the ***TURNOVER® for iSeries User Guide***.

Enroll programmers - Main Menu option **2** ❑
See ***Chapter 2: Setting up Programmer Profiles*** in the ***TURNOVER® for iSeries User Guide***.

Define the computers in your network - Main Menu option **11**, Option **1** ❑
Install TURNOVER® for iSeries on your remote systems, if you can. See ***Chapter 1: Preparing to Install, Upgrade, or Apply Changes*** in the ***Getting Started with TURNOVER® for iSeries*** guide.

Also, do **Part III** of this checklist.

Copy the application model(s) that best suits your needs. ❑
**In the Client:** Right-click an application definition and choose **Copy Application**.
**On the IBM i: Main Menu option 1, option 3**
See ***CHAPTER 3:*** *SETTING UP* in this guide.

Modify your custom model to match your requirements. ❑
**In the Client:** Right-click an application definition and choose **Checkout Application**.
**On the IBM i:** Main Menu option 1, option 2.

Now create your application definition by copying your model. ❑
**In the Client:** Right-click your custom model and choose **Copy application**.
**On the IBM i:** Main Menu option 1, then option 3.
See Method 1: Copying an Application Model on page 4-2 in this guide.

Define the TURNOVER® for iSeries project(s) you will need. ❑
Use TURNOVER® for iSeries Client Projects, or
Main Menu option 9, then F6.
See the ***TURNOVER® for iSeries Client User Guide*** or
***Chapter 9: Working with Projects and Tasks*** in the ***TURNOVER® for iSeries User Guide***.

Create test objects with which to test your setup. ❑

Test each object type using your application definition. ❑
Create a task, build a worklist, add test objects to the worklist, check out objects, build and run TURNOVER® for iSeries forms, distribute, and install changes (if appropriate).

[10] Remember, Main Menu options correspond to chapters in the ***TURNOVER® for iSeries User Guide***.

# DISTRIBUTION SETUP CHECKLIST

You might find it helpful to read the Overview of distribution in ***Chapter 11*** of the ***TURNOVER® for iSeries User Guide***.

**On the Development computer:**

If you haven't already, define all the TURNOVER® for iSeries computers in your network, including the development computer. ❑

Define the Send distribution defaults for each production computer. ❑

**In the Client:** Change a Distribution System and go to the Send Defaults page.
**On the IBM i:** Main Menu 11, Option 1; then select a system with option 9.
(If any type of "source" is to be distributed for any application, you must enter a **Y** in the distribute source field in that computer's Send defaults. You will be able to restrict source distribution by application or by object type, if desired. )

Verify the operating system levels of all computers involved and set the **TGTRLS** global default value on the development computer accordingly. (You can override this value for a specific application if necessary.)

**In the Client:** Expand Administration and then Utilities, then double-click Global Defaults. The Target Release parameter is on the Form-Related page.
**On the IBM i:** Main menu option 8, then option 5

Verify that the message receiver autostart job (TOMSGRCV) is running in subsystem. ❑

Autostart job TOMSGRCV must be running on the development system. This job is set up for you during installation. It waits for a confirmation message to be sent from the production computers, and then updates the form status there.

Create a remote application definition for each distributing application level, and distribute it to each production computer. ❑

(Do this now only if you have both created your application definitions AND set up your distribution defaults. If you don't do this now, you can come back and do it later.

**On Each Production Computer:**

Create system definitions for the development computer and the production computer. ❑

**In the Client:** Expand Administration, then right-click Distribution Systems and choose New System.
**On the IBM i:** Main Menu option 11, then option 1.

Define the Receive distribution defaults for the local production system. ❑

The target user must be the same as that defined on the development system.

Verify that the TURCRARE autostart job is running in subsystem TSERVER. ❑

## Appendix A: TURNOVER® for iSeries Configuration Checklists

This job is set up and started for you in subsystem TSERVER during installation. The TURCRARE autostart job must be running on each production system. This job waits for a distribution to be sent to the target user, and then receives the distribution file and checks the RECEIVE distribution defaults to determine if the form should be submitted.

Review the global type codes on each production system. ❑

Have you created any additional global type codes on your development system? You must create them on your production systems as well.

Review the application definition on each production system. ❑

In your remote application definition, you will probably want to change **Method** to **MO** if you are distributing objects only, ***Require source checkout*** to **N**, and ***Require project entry*** to **N**.

If you have any questions, contact your local Technical Support Representative as outlined on the SoftLanding website.

# Index



## Index