SoftLanding SYSTEMS
A Division of the UNICOM Group of Companies

UNICOM SYSTEMS, INC.

Supplement #42

84 Elm Street • Peterborough, NH 03458 USA
TEL (010)1-603-924-8818 • FAX (010)1-603-924-6348
Website: http://www.softlanding.com Email: techsupport@softlanding.com

MIGRATING FROM ADM TO TURNOVER® FOR ISERIES V100

This document describes how to migrate from IBM's Application Development Manager (ADM) to UNICOM Systems, Inc.'s TURNOVER® for iSeries v100 change management application. The document assumes that you are familiar with ADM, and it contains these main sections:

INTRODUCTION TO TURNOVER® FOR ISERIES V100

TURNOVER® for iSeries v100's strength is its ability to handle virtually any kind of application development with a single process. As iSeries applications become more modular and begin to include components built with non-traditional technologies (such as Java, XML, C++, and graphics), this unified approach to application development lets IT organizations spend more time designing and improving competitive applications, and less time managing them.

In addition to supporting basically any development technology, TURNOVER® for iSeries v100 also manages modifications to packaged applications such as PeopleSoft® World (formerly J.D. Edwards WorldSoftware) and BPCS, as well as applications that are built using development tools such as LANSA/LANSA for the Web, AS/SET, COOL:Plex, and COOL:2E.

As the original project-driven change management system, TURNOVER® for iSeries v100's evolution has paralleled trends in AS/400 and iSeries application management. From its foundation of reliable source and object control, it has grown to support automated promotions, auditable change history for source and objects, promotion recovery, application deployment to remote systems, version control, project management, and more. TURNOVER® for iSeries v100's modularity makes it both easy to implement and easy to learn. Individual users can choose from its standard 5250 user interface or its native Windows graphical user interface.

TURNOVER® for iSeries v100's evolution has also been tightly synchronized with advancements in AS/400 and iSeries technology. It was the first change management product to:

- Fully support ILE (concurrent with V2R3 general availability);
- Manage Java and other objects stored in the IFS (May 1999);
- Offer TCP/IP support (October 1997) and FTP support (September 2001);
- Fully automate WebSphere application management and deployment (August 2001); and
- Integrate with the Eclipse integrated development environment (IDE) through its PVCS plug-in (April 2002).

UNICOM Systems, Inc.'s status as the only change management member of IBM's exclusive System Management Partner Group helps ensure continued rapid support of all key technology advancements.

Just as importantly, TURNOVER® for iSeries v100's technology is backed by complete implementation and training services, online workshops for continued education, and the industry's most responsive technical support.

COMPARISON OF TURNOVER® FOR ISERIES V100 AND ADM

This section provides a comparison of ADM and TURNOVER® for iSeries v100 terminology and features.

Glossary of terms

This is a list of ADM terms with their TURNOVER® for iSeries v100 equivalent.

ADM Term	**TURNOVER® for iSeries v100 Equivalent**
Project	Application
Group	Application Level
Parent Group	Target Level
Part	"Native" OS/400 Term: Member, Object, Message ID, and so on
Part Type	Type Code
Build Option	Create Command (**CRTRPGPGM**, **CRTBNDRPG**, and so on)
Build Option	Pre/Post Run Command (**OVRDBF**, **CRTDUPOBJ**, and so on)
Part List	Form
Promote Code	Highest Unlocked Level
Search Path	Library List Object
Access key	Checkout record

Of the above, the TURNOVER® for iSeries v100 concepts that probably need the most explanation are *application* and *application level*. In ADM, you have projects and groups; in TURNOVER® for iSeries v100, you have applications and levels. In TURNOVER® for iSeries v100, an application consists of an identifier, and the definition itself. The application identifier is a three-part "code" consisting of a four-character name code, a two-digit release code, and a two-digit version code. For example, our application for TURNOVER® for iSeries v100 is TO/10/00. (It is common to leave the release and version fields as zero.)

An application has levels, which are identified using a two-digit level number at the end of the application identifier (as in TO/10/00/05). The sequence of the level numbers from lowest to highest represents the promotion steps from Development to Production.

The definition portion of a TURNOVER® for iSeries v100 application consists of a series of settings that are grouped into categories. Each application level has its own definition. A two-level application is typical, where the first level describes the promotion from Development to QA and the second from QA to Production. You can essentially have any number of levels you need to satisfy your requirements. The promotion path is a linear process from the lowest to the highest.

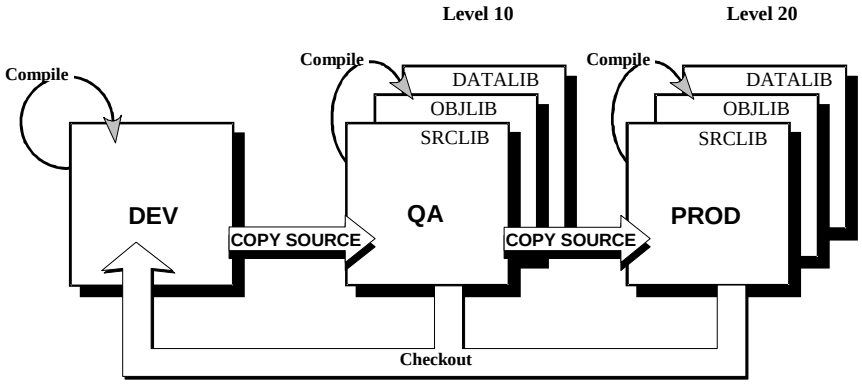

Figure 1: Common two-level TURNOVER® for iSeries v100 application

If you modify vendor software or work on multiple releases of software, you would use what we call "locked levels." Locked levels are assigned higher numbers than the other levels. Consequently the "Production" level is often referred to as the "highest unlocked level." When you have a locked level, you can check out source of that level (if it has not been previously modified), but you can promote back only as far your highest unlocked level. This effectively keeps your modifications separate from the "base."

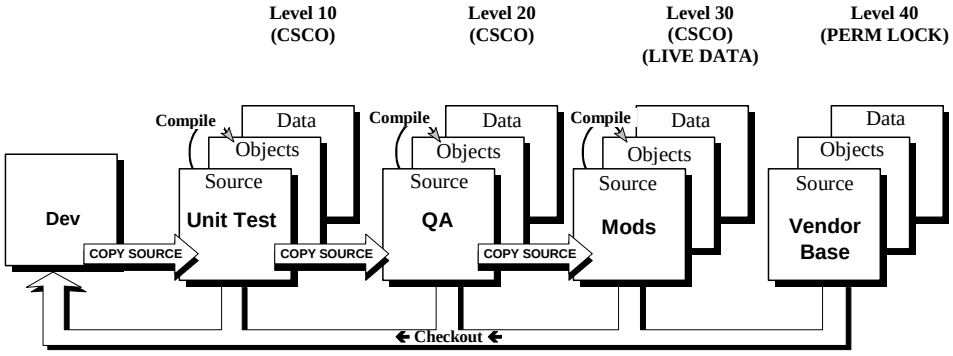

Figure 2: TURNOVER® for iSeries v100 application with a permanently locked level

Conceptually, ADM does something similar using groups and promote codes.

One significant difference between TURNOVER® for iSeries v100 and ADM is that TURNOVER® for iSeries v100 applications simply lie over the top of your existing libraries, whereas ADM creates its own internal libraries between which that you must import/export.

Another important difference is that TURNOVER® for iSeries v100 treats an object and its source as a single entity, comprised of a source member and an object of a given object type. ADM treats an object and its source as two distinct parts. With TURNOVER® for iSeries v100, you can perform a checkout by either the object name/type or the source member, and both are effectively checked out for you. Likewise, they are promoted back to production together. TURNOVER® for iSeries v100 always treats them as a unit.

Additional terms

Here are the definitions of some additional terms that are commonly used in TURNOVER® for iSeries v100:

Form You handle TURNOVER® for iSeries v100 promotions by first creating and then running a TURNOVER® for iSeries v100 *form*. Each form has a unique number that acts as a TURNOVER® for iSeries v100 job number. Although forms are similar to ADM part lists, part lists are optional in ADM. Forms are an essential part of TURNOVER® for iSeries v100 because TURNOVER® for iSeries v100 derives much of its power and ease-of-use through this concept. For example, forms drive TURNOVER® for iSeries v100's ability to automatically recover failed promotions, provide fast access to change history, and track and coordinate changes across all of your systems.

Project TURNOVER® for iSeries v100 uses the term *project* in the traditional project management sense. TURNOVER® for iSeries v100's integrated project management system includes both Helpdesk and Call logging facilities. A project is a grouping of related tasks, and is identified with a four-character project name. For example, APCR might be the project name for your Accounts Payable Change Requests project.

Task *Tasks* are created within projects. A task is where all information for a specific request is stored and tracked. Task identifiers are typically sequentially assigned, and consist of four alphanumeric characters. Task numbering can be simple (numerals only) or aggressive (numerals and letters).

Subtask Tasks can also have *subtasks*, which are the same as tasks, except that they are grouped beneath the task using a two-digit subtask identifier.

Reference All changes in TURNOVER® for iSeries v100 are assigned and grouped by their project *reference*. The project reference is a ten-character code that is typically a combination of the project, task, and subtask identifiers (as in APCR000103). The project reference lets you easily organize your work around a specific assignment, and becomes a permanent audit trail between a specific revision of an item, and its originating request.

Worklist Another core TURNOVER® for iSeries v100 feature is the Programmer Worklist Manager (PWM), also called "the worklist." The worklist provides a PDM-style interface to the edit/compile/debug process, but integrates this with all of its CM and project management functions. One of the strengths of the worklist concept is that you can easily switch between worklists. The worklist ID corresponds to the project reference (change request) with which it's associated. Also, all item types can be on one worklist, saving you, the programmer, from having to jump between source files and allowing you to see only the items on which you are working for that specific change. You can configure your worklists to use Code/400 in place of SEU/SDA.

X-Ref TURNOVER® for iSeries v100 comes with a complete cross-reference database, and maintains it automatically as program changes are made. TURNOVER® for iSeries v100 uses this information to recreate dependent objects automatically when a change is made to objects such as database files or ILE modules.

Distribution TURNOVER® for iSeries v100 has the built-in ability to distribute objects or source, or both, to any number of iSeries servers, which we refer to as "Remote" or "Production" computers. These items can then be installed on that system using the standard TURNOVER® for iSeries v100 form promotion process. Installations can be scheduled and synchronized across all production systems, and a failure on one can trigger a rollback on all.

Key differences between TURNOVER® for iSeries v100 and ADM

This topic describes the differences between the way TURNOVER® for iSeries v100 and ADM implement change management concepts.

Library structure

We have discussed this previously, but it bears repeating. When defining an ADM project group, ADM assigns library names based on the short project name and short group name. You perform your development work is then within this library structure and promote it within the library structure. At different points within the development cycle, you would export related parts to a set of libraries either for testing or for production use.

In contrast, the TURNOVER® for iSeries v100 process builds a TURNOVER® for iSeries v100 application over your production, testing and, development libraries (and others as needed). When you check out an object, TURNOVER® for iSeries v100 copies its source from production into the development library (see **Figure 1: Common two-level TURNOVER® for iSeries v100 application** on page 4). After you make changes in development, you create and run a TURNOVER® for iSeries v100 form, which moves the selected objects and source from your development library into your testing libraries. After you complete testing, you run and copy this form, moving the objects and source from test into production, where TURNOVER® for iSeries v100 then checks the object back in.

Import

In ADM, at some point you import the objects or source, or both, into the ADM project.

With TURNOVER® for iSeries v100, this is not necessary. Because TURNOVER® for iSeries v100 builds the application definition over your existing libraries, the objects are already under TURNOVER® for iSeries v100's control. All that is needed is an "initial build," which TURNOVER® for iSeries v100 uses to populate the initial audit information about your objects so it can later tell if an object was changed "outside of TURNOVER® for iSeries v100."

Export

In ADM, after you complete development work on a part or group of parts, you promote the parts to the next group. You might also export the parts to your own libraries for testing.

With TURNOVER® for iSeries v100, the process is a bit different. Because TURNOVER® for iSeries v100 builds the application definition over your existing libraries, you only need to promote objects. You do this by running a form, which moves the objects from one level to the next. Once the form has run, the objects are available in the target libraries for use. During your migration to TURNOVER® for iSeries v100, you may have to export parts from the ADM *project.group* library structure into your own libraries.

Promotion

With ADM, promotion is a multi-step process. After you complete development work on the source, you promote the source to the next group. You then build the objects into that group, and, if necessary, export those objects into your own libraries for testing or production use.

With TURNOVER® for iSeries v100, the TURNOVER® for iSeries v100 form process controls the promotion of objects. You add the required objects to the form. TURNOVER® for iSeries v100 determines if additional objects should be added, such as a logical file or programs when a physical file is on the form, by referring to its integrated cross-reference database. When you run the form, TURNOVER® for iSeries v100 archives any existing objects and source (if you have told it to), copies the new source to the target library, and compiles the objects.

Check-in

TURNOVER® for iSeries v100 has a check-in option, but it has a different meaning than in ADM. In TURNOVER® for iSeries v100, you only use the check-in option to cancel a previous checkout. Otherwise, check-in occurs automatically when items are promoted back to the location from which they were checked out. They are then available for checkout again.

Archiving

In ADM, you can have up to five previous source versions archived. Archiving is based on the member name of the source. Therefore, if you have a member MYCMD in QCMDSRC, and a member MYCMD in QCLSRC, archiving of these go to the same ADM archive, possibly overlaying the source of the other. Objects are not archived in ADM.

TURNOVER® for iSeries v100's archiving features are more robust. Each TURNOVER® for iSeries v100 application level can have its own source archiving file, but application levels can also share archiving files with other application levels. When objects are promoted to the next level, existing source code is moved to the archive file for that application level. There is no limit on the number of versions archived for a particular source.

With TURNOVER® for iSeries v100, archiving is not member name specific. When source is moved to the archive file specified for that application level, the member is given a unique name, and that name is recorded within TURNOVER® for iSeries v100. Through the TURNOVER® for iSeries v100 object history, you can easily view previous versions of your source members.

TURNOVER® for iSeries v100 not only archives source; it also archives objects. During a TURNOVER® for iSeries v100 form run, any existing objects are moved to an archive library. Each archive library is unique to the particular form run. (For more information, see ***Reversing changes*** on page 13.)

Reason control

ADM provides you with a part-list part to aid in reason control. This part consists of a list of related parts.

With TURNOVER® for iSeries v100, the TURNOVER® for iSeries v100 form shows you each object that was promoted for a given change request. In addition, the TURNOVER® for iSeries v100 project system allows you to track changes from the initial user request, through project completion. As a developer, you can enter free-format text into your assigned tasks to detail the changes you have made. TURNOVER® for iSeries v100's Programmer Worklist Manager groups your projects into manageable pieces, separating the work for one project from that of another. You can also record the time you have worked on each project. Finally, there are rich reporting options for all of the above features.

File changes

TURNOVER® for iSeries v100 provides robust support for making database changes. TURNOVER® for iSeries v100 automatically rebuilds logical files, recreates members, copies data, restarts journaling, reapplies triggers and constraints, and so on. It also provides full support for database objects created via DDS or SQL DDL.

TURNOVER® for iSeries v100 has very flexible data copy options. You can:

- Use ***MAP**, ***DROP**, or ***NOCHK** to have TURNOVER® for iSeries v100 copy data back.
- Use ***NOCOPY** to have TURNOVER® for iSeries v100 ignore data.
- Use ***FROMFILE** to have TURNOVER® for iSeries v100 copy data forward with the file.
- Configure data to call a custom user-written program to perform the data conversion and copy.

With our optional TURNOVER® for PDQ v100 product, you can install database changes without the application being off line for more than a few minutes, as opposed to typically needing to be down for the entire time it takes to copy data and rebuild access paths.

TURNOVER® for iSeries v100 handles database file changes hand-in-hand with program changes and provides sophisticated data file support.

Code/400 integration

Like ADM, TURNOVER® for iSeries v100 allows you to use Code/400 as the source editor/screen designer. However, ADM's integration with Code/400 is more extensive at this point. Once Code/400 is rebuilt over IBM's new IDE, TURNOVER® for iSeries v100 will offer complete integration. (For more information, see ***TURNOVER® for iSeries v100 Support for Java, IFS, and the*** New IBM Workbench IDE on page 15.)

Build

ADM has a powerful build feature, offered as a single option, that uses information obtained "on the fly" from the compiler to determine what to rebuild and when. TURNOVER® for iSeries v100 has similar features, but it works differently. TURNOVER® for iSeries v100 provides this feature during the promotion process by interrogating its cross-reference database and adding related items to the form for recompiling. You can do this automatically, or you can do manually if you want to control which objects are rebuilt.

TURNOVER® for iSeries v100, unlike ADM, does not recompile related objects automatically while you are working in development. Instead, you can select an option to bring up a panel of "related objects" from the cross-reference database, and then choose an option to compile those objects in development. Alternatively, you can do this automatically in TURNOVER® for iSeries v100 by using a "Unit Test" level as the first stage of the application. You would build and run a form to move the objects into the Unit Test area, and standard form processing would recompile the related objects along the way.

Performing steps in TURNOVER® for iSeries v100 that you did in ADM

Most ADM options have a one-to-one mapping to TURNOVER® for iSeries v100, and will be very easy for you to reconcile. We intend to research this issue further and provide documentation that details the steps you should follow for any option, such as the ADM Build option, where the differences between ADM and TURNOVER® for iSeries v100 are greater.

TURNOVER® for iSeries v100 features that are not available in ADM

Projects, Helpdesk, and Wisedesk

TURNOVER® for iSeries v100 includes a complete project management system, including Helpdesk and Call logging functionality. As was noted previously, the project-oriented focus of TURNOVER® for iSeries v100 appears throughout the product and ties together many of its features. Wisedesk is a powerful facility for organizing and locating tasks that are in the project system. You can use this to automatically categorize information in any structure that you want, allowing easy retrieval when you are researching users' questions.

The project system's automated messaging and workflow are also tied into the change management process. As a simple example:

- When you are assigned a task, TURNOVER® for iSeries v100 can send you a notification email.
- When you promote changes to QA, TURNOVER® for iSeries v100 can send a notification email to the QA manager.
- When you promote changes to production, TURNOVER® for iSeries v100 can send a completion email to the requester.

Object history

TURNOVER® for iSeries v100 maintains a detailed object history database that you can easily access using online panels and reports. Not only can you see all of the times that a given object has been modified, but you can also locate the change requests that initiated the change and browse the archived source associated with it. All of these features are very powerful problem-solving tools.

Approvals

You can attach an approval process to forms. You can use this feature to ensure that a form cannot be promoted to production without, for example, the approval of the QA resource. This can all be tied into the larger project management and messaging workflow.

Emergency changes

TURNOVER® for iSeries v100's emergency change procedures allow you to check out an item for an emergency change, even if it is currently checked out for a normal change. The emergency change can follow an alternate – and usually shorter – path back to production. Administrators and programmers are automatically messaged to ensure that everyone is aware of the emergency change. TURNOVER® for iSeries v100 prompts the programmer who already has the item checked out to incorporate the emergency change before promoting back to production. Source compare and merge features are available on the Programmer Worklist to ease this effort.

Authority to the emergency change process is easily controlled and limited. Additional alerts can be created to ensure that people know an emergency change has been made.

Related applications

TURNOVER® for iSeries v100 offers a feature called "related applications" for handling the simultaneous development of multiple releases of the same software. Suppose you check out a program in Release 5.0. TURNOVER® for iSeries v100 automatically informs you that the item has been modified in Release 5.2 and 5.3, and assists you in making the same change in all of these releases. You cannot promote the change in the base release without either making the change in the later releases or explaining why the change does not need to be made in those versions.

Synchronizer

TURNOVER® for iSeries v100 includes a full vendor release management subsystem called Synchronizer. This utility provides tools to automate as much as possible the merging of a new vendor release with custom modifications that have been made in-house. We have many documented cases where this has been used to dramatically reduce the time it takes a customer to bring up a new vendor release.

Reversing changes

As long as you are archiving objects, you can quickly and easily back out a successful promotion (even across multiple systems) by creating and running a *Recovery Form*. This restores the previous versions of the source and object associated with the change. You can even roll back only selected objects, which might be desirable when a big change was installed but only one or two objects have problems. Either way, once the problem is resolved, you can apply new fixes using standard TURNOVER® for iSeries v100 procedures.

ILE conversion

TURNOVER® for iSeries v100 has a built-in feature that automates the conversion of RPG to ILE as part of the normal change process. Technically, you can use it to convert any type of source to any other type, as long as you can provide the conversion utility.

Support for IFS (Java, HTML, WebSphere …)

TURNOVER® for iSeries v100 has complete support for IFS objects in the base product. No additional purchases are required. This is covered in the ***TURNOVER® for iSeries v100 Support for Java, IFS, and the*** New IBM Workbench IDE topic on page 15.

Interfaces (JDE, LANSA, 2E, Plex, AS/SET, and ProGen)

TURNOVER® for iSeries v100 provides interfaces to many tools and applications that require special processing, such as J.D. Edwards, LANSA, Cool:2E, COOL:Plex, AS/SET, and ProGen. In all cases, you can use TURNOVER® for iSeries v100 the same way you handle your "native" programming tasks, enjoying full CM support while still retaining all of the features and benefits your tool provides.

Instead of its native TURNOVER® for iSeries v100 database, TURNOVER® for iSeries v100 can also use the cross-reference databases built by Hawkeye's Pathfinder or ASC's Abstract.

Exits, APIs, and customizations

UNICOM Systems, Inc.'s commitment to providing an open system is long standing. To that end, TURNOVER® for iSeries v100 includes an extensive list of exit points. These exit points insert calls for customer-written programs into the standard TURNOVER® for iSeries v100 change process, letting you add function that is not supported natively. For example, if a scheduled promotion fails, you might want to be paged – or maybe you require special messaging, such as sending notifications to a mainframe.

TURNOVER® for iSeries v100 provides APIs in the form of OS/400 commands for just about every function. This makes it easy for you to write exit programs, or to use PDM user options or similar features in other tools.

There are many other features that we have not listed here that are available in TURNOVER® for iSeries v100 but not in ADM. We encourage you to review the TURNOVER® for iSeries v100 product brochures, or speak with a UNICOM Systems, Inc. representative. We will gladly answer your questions.

TURNOVER® FOR ISERIES V100 SUPPORT FOR JAVA, IFS, AND THE NEW IBM WORKBENCH IDE

Support for the Integrated File System (IFS) and Java was added to TURNOVER® with Release 5.1 in 1999. Knowing that virtually all our customers would eventually need IFS support, we chose to integrate it as a standard part of TURNOVER® rather than create a stand-alone product that customers would purchase separately. This approach helped us achieve a significant installed base of customers using TURNOVER® to manage objects in the IFS.

Once we had the core CM engine and APIs for IFS in place, we moved on to build a Windows-based graphical user interface to TURNOVER®'s IFS support. The GUI, called TurnOver eCM, became generally available in May 2000 as part of TURNOVER® Release 5.2. Because TurnOver eCM was built as a rich GUI interface instead of a separate product, its core processing power comes from TURNOVER® on the iSeries. This architecture allows users to accomplish virtually anything that the iSeries itself supports through the IFS, and makes it possible for TURNOVER® to support advanced deployment and promotion options. The TURNOVER® for iSeries v100 Client user interface has subsequently replaced the TurnOver eCM GUI.

TURNOVER® for iSeries v100, the IFS, and deployment

You can use TURNOVER® for iSeries v100 to manage files that reside on NT/2000 servers using the /QNTC file system; Novell Netware servers through the /QNETWARE file system; and Unix/Linux servers using the Network File System (NFS). TURNOVER® v5.3, released in September 2001, includes the ability to deploy files using FTP. TURNOVER® for iSeries v100 deploys files to any type of server that supports FTP, as well as to locations outside of the corporate firewall and LAN. UNICOM Systems, Inc. uses this feature internally to manage its public Web site (http://www.softlanding.com), which resides at an external hosting facility. For companies working with client/server applications, TURNOVER® for iSeries v100 integrates with Deploy/PC, a separate product that installs updates on Windows desktop computers, without user intervention.

TURNOVER® for iSeries v100, the IFS, and promotions

Because support for the IFS is integrated into TURNOVER® for iSeries v100, you can promote all native iSeries objects, such as DB2/400 objects and RPG programs, in the same job as related IFS objects, such as web content. TURNOVER® for iSeries v100 synchronizes the promotions and you can easily recover them if a problem or processing error occurs, or roll them back later. TURNOVER® for iSeries v100 also lets you run commands as part of the promotion process. For example, if you are working with Java, you can have TURNOVER® for iSeries v100 automatically invoke the **CRTJVAPGM** command after you promote a class or Jar file.

PC version control

TURNOVER® for iSeries v100's version control for IFS objects has a definite slant towards the iSeries way of doing things. However, most of the source and files stored in the IFS are created on a developer's PC. Understanding that web/Java/PC programmers might be more comfortable using PC-based version control at the development level, we introduced TURNOVER® for SVN v100 to give our customers a choice of using iSeries-based or PC-based PC version control. TURNOVER® for SVN v100 is widely used version control product in the market, now integrates tightly with TURNOVER® for iSeries v100. The combination of TURNOVER® for iSeries v100 and TURNOVER® for SVN v100 allows us to offer a true "Best of Breed" solution to the iSeries marketplace.

WebSphere/Eclipse IDE support

TURNOVER® for iSeries v100 holds the unique position of providing full lifecycle support for WebSphere application development today. Because of MERANT's leadership position in the Eclipse project, PVCS Version Manager is already integrated into the currently-shipping WebSphere Studio Application Developer (WSAD) and soon-to-be-shipping WebSphere Studio Site Developer (WSSD). Based on the integration of PVCS with WSAD, and the TURNOVER® for iSeries v100 integration with PVCS, TURNOVER® for iSeries v100 can provide complete support from "development through deployment" to companies that are using the new IDE to build their Web applications. TURNOVER® for iSeries v100 currently provides RPG support through its native interface to Code/400, and it will provide additional support in the future as IBM integrates the functionality that now exists in Code/400 and related tools into its new IDE.

UNICOM Systems, Inc. has also been on the leading edge with WebSphere Application Server (WAS) on the iSeries. In 2001, we developed a set of tools that let you manage your WAS servers using OS/400 commands. We released these commands under an Open Source license agreement (GPL). IGNITe/400 hosts the project on its web site (ignite400.org). Largely based on this project, TURNOVER® for iSeries v100 can control WebSphere during its promotion process. For example, if you are promoting a change to Java code that requires stopping and restarting the application server, you can do this automatically within the TURNOVER® for iSeries v100 promotion. More advanced users can even use TURNOVER® for iSeries v100 to promote and install changes to the WAS configuration as part of the same process, such as when a new Servlet definition needs to be added to the Web Application.

Change management your way

Finally, we cannot resist discussing TURNOVER® for iSeries v100's extraordinary flexibility and the opportunities it offers you to modify and control your own change management processes. For example, in Release 5.3, we added the ability for you to define your own "Change Management Schemes" and then attach these schemes to types that are being managed by TURNOVER®. This lets you do things like build your own custom schemes to manage the records in a database file – with full and complete CM functionality.

This makes it very easy for you to quickly extend the TURNOVER® for iSeries v100 product in ways that we cannot imagine today. Recently, one customer implemented a strict new set of audit requirements, whereby the creation date and time of objects must match on all of their systems. Normally, when you use TURNOVER® for iSeries v100 to distribute and install objects on other iSeries servers, there is a step in the process where the existing object is duplicated, via **CRTDUPOBJ**, so that it can be sent to the other system. This duplication causes the new object to have a different creation date and time. We were able to meet the customer's needs by creating a new CM Scheme that overrides our use of the **CRTDUPOBJ** command to use save and restore commands, thus preserving the creation date and time.

There are a lot of reasons we would not want our "base" product to use this technique. However, by giving you the ability to easily override TURNOVER® for iSeries v100's standard behavior, you can customize the product to your own requirements.

PROCEDURES FOR MIGRATING FROM ADM TO TURNOVER® FOR ISERIES V100

This section explains the basic procedures for migrating from ADM to TURNOVER® for iSeries v100. The procedures listed here may vary, depending on your environment and the specific needs of your company. UNICOM Systems, Inc. provides on-site training and services to aid in your migration. To avoid migrating several groups for each project, you should migrate from ADM to TURNOVER® for iSeries v100 once all objects have been checked in and promoted to the root group.

Prerequisites

To use the ADM migration tools, you must:

- Have TURNOVER® for iSeries v100 installed on your system.
- Know how to set up a TURNOVER® for iSeries v100 application.
- Know your current ADM environment.
- Have Deploy/400 installed.[1]

General overview

These are the general steps you will follow when migrating an application from ADM to TURNOVER® for iSeries v100. You will not need to do all of these steps each time, and some of the steps are optional, depending on your migration requirements.

1) Create TURNOVER® for iSeries v100 applications and levels.

2) Define developers to TURNOVER® for iSeries v100 and grant them authority.

3) Determine which ADM part types you are currently using.

    Map TURNOVER® for iSeries v100 type codes to part types.

4) Map ADM project groups to TURNOVER® for iSeries v100 application levels.

[1] We deliver the ADM migration tools as a Deploy/400 package. Therefore, to install the migration tools, you must have Deploy/400 installed (available on the TURNOVER® for iSeries v100 installation CD). For Deploy/400 installation instructions, read "***Preparing to Download and Apply TURNOVER® for iSeries v100 Fixes and Quarterly Updates***" in ***Chapter 5: Additional Installation Topics*** of ***the Getting Started with TURNOVER® for iSeries v100*** guide.

5) Customize creation commands.

    Steps for Customizing Creation Commands

6) Export parts.

    Select parts for export.

    Run export job.

7) Perform TURNOVER® for iSeries v100 initial build.

8) Assign pre- and post-run commands.

9) Review work in progress.

These steps are described in detail on the following pages.

1) Create TURNOVER® for iSeries v100 applications and levels

When defining an ADM project group, ADM assigns library names based on the short project name and short group name. You perform and promote your development work within this library structure. At different points within the development cycle, you export related parts to another set of libraries, either for testing or for production use.

The TURNOVER® for iSeries v100 process differs from this by building a TURNOVER® for iSeries v100 application over your production, testing, and development libraries (and others as needed). Therefore,

- When you check out an object, TURNOVER® for iSeries v100 copies its source from production into the development library.
- After you make changes in development, you create and run a TURNOVER® for iSeries v100 form to move the selected objects and source from your development library into your testing libraries.
- After you complete testing, you copy the form and run it to move the objects and source from test into production, where TURNOVER® for iSeries v100 checks the object back in.

At this point, you need to analyze your current ADM projects and groups, determine what TURNOVER® for iSeries v100 application structure is appropriate for your environment, and create all necessary applications and levels. For extensive guidance in doing this, see the ***TURNOVER® for iSeries v100 Application Planning Guide***. Also, UNICOM Systems, Inc. can provide on-site training to aid you in this process. (We highly recommend this.)

UNICOM Systems, Inc. Recommends

You can select type codes for your application at this time, but it is not necessary that all required type codes be added to the application at this point. TURNOVER® for iSeries v100 type codes are addressed in more depth in steps 3) and 5) and we recommend that you return to your application definition at a later time to ensure that the necessary compliment of type codes exist.

2) Define developers to TURNOVER® for iSeries v100 and grant them authority

You must enroll your developers in the TURNOVER® for iSeries v100 system. On the TURNOVER® for iSeries v100 Main Menu, choose option **2** (***Maintain programmer information***) to define basic information about each developer, such as the development libraries s/he uses and to which output queue his/her output should go.

Also, you need to authorize each developer to the applications s/he will use. On the TURNOVER® for iSeries v100 Main Menu, choose option **3** (***Maintain TURNOVER® for iSeries v100 Authority***) to define each user's TURNOVER® for iSeries v100 authority. (You do this process once, during initial user setup.)

You can use TOADMUTIL menu option **3** (***Work with project/application cross reference***) to easily see which users are enrolled to which ADM projects and then selecting a project/group with option **5**. This shows you all users enrolled in the selected project, their enrollment type (***ADMIN** or ***DEVELOPER**), and the developers' access levels to the project groups (***READ** or ***UPDATE**). Use this information to determine who should be enrolled as a developer in TURNOVER® for iSeries v100 and to which applications they should be authorized.

3) Determine which ADM part types you are currently using

To help configure TURNOVER® for iSeries v100, you need to analyze what ADM part types you are currently supporting. To do so, use the TOADMUTIL menu options as follows:

- Use menu option **1** (***Generate Part Languages***) to examine the ADM parts file (QALYPART1). This lets you determine which parts and languages you are using in your ADM environment and how many parts there are of each combination. When you run the command, it obtains a snapshot of your ADM system. (If you add or delete objects from ADM, you can re-run the command to update the counts.)
- Use menu option **2** (***Work with Part Mappings***) to see the results of the Generate Part Languages menu option (see above). You can see the results for a specific ADM project or for all ADM projects at once.

3.1) Map TURNOVER® for iSeries v100 type codes to part types

Use menu option **2** (***Work with Part Mappings***) to map a TURNOVER® for iSeries v100 type code to each Part/Language combination. These type codes are used in Step **6)** ***Export parts*** (see page 25).

Typically, because TURNOVER® for iSeries v100 type codes for parts are usually the same for each project, you will filter the panel to show ***ALL** projects. However, if there are parts in projects that do not conform, you can map a different type code to those individual projects.

The most common TURNOVER® for iSeries v100 type codes for each part type supported are pre-defined. Therefore, normally you just need to review the list for parts that you manage with ADM and verify that a type code is mapped to those parts. If you manage parts through ADM that do not have a mapped TURNOVER® for iSeries v100 type code, you must manually select the appropriate TURNOVER® for iSeries v100 type code or, if one is not already defined, create one and select it.

4) Map ADM project groups to TURNOVER® for iSeries v100 application levels

To aid the migration from ADM to TURNOVER® for iSeries v100, you must map your ADM project groups to the corresponding TURNOVER® for iSeries v100 application levels. The information gathered by this step is used later in the migration process. This step saves you from having to provide the same information every time.

Use TOADMUTIL menu option **3** (***Work with Projects/Applications cross reference***) to list all your ADM project groups. You can then map the ADM project groups to the corresponding TURNOVER® for iSeries v100 application levels.

If the ADM group you are mapping is used for development, set the ***Development*** flag to **Y**. If you are using *PGMR libraries in your TURNOVER® for iSeries v100 application, you can specify which programmer is to be mapped to that group.

5) Customize creation commands

Overview of TURNOVER® for iSeries v100 processing

When you check out and promote objects through TURNOVER® for iSeries v100, the TURNOVER® for iSeries v100 type code controls their creation. The type code describes how TURNOVER® for iSeries v100 should compile an object and it also defines the "from" and target libraries for that object type.

TURNOVER® for iSeries v100 gives you a number of methods for customizing creation commands. You start with TURNOVER® for iSeries v100 Main Menu option **8** (***Utility menu***), followed by option **4** (***Work with type code definitions***). The global type code definition panel appears. If you look at a type code such as RPGLE (by selecting it with option **2**, ***Change***), you will see the creation command, with the many substitution variables that are available. By setting different values for these substitution variables, for each TURNOVER® for iSeries v100 application level, you can control how objects are created (compiled) at that level.

Now select TURNOVER® for iSeries v100 Main Menu option **1** (***Work with application definitions***). To see the type codes defined for an application level, select the application with option **5** (***View***) and select the level with option **14** (***Type codes***). Press **F8** (***User-defined parameters***) to see the values associated with each variable for that level of the application.

You should now see, by setting different values for these substitution variables, how you control how TURNOVER® for iSeries v100 compiles an object at every application level. For example, when you promote an RPGLE program into your test environment, you might want to specify **DBGVIEW(*SOURCE)**. However, when promoting into production, you would use **DBGVIEW(*NONE)**.

Overview of ADM processing

In ADM, you customize creation commands using a Build Option part. When you build a part within ADM, ADM uses the following hierarchy to determine which Build Option (BLDOPT) part to use:

1) Use a BLDOPT part with the same name as the part name being built.

2) Use a BLDOPT part with the name of the default compiler.

3) Use a BLDOPT part with the name of QDFT.

Once ADM finds the BLDOPT part, it looks for a label with the same name as the part type being promoted. If ADM finds a label, it runs the CL command on that statement. If ADM does not find a label for the part type, it retrieves the default command for that BLDOPT part and it uses that customized command. If ADM does not find the default command, it uses the default command for that part type without customization.

5.1) Steps for customizing creation commands

a. Use the TURNOVER® for iSeries v100 **TDSPBLDOPT** (**Display Build Options**) command to view the commands within a BLDOPT part.

b. On the ***Work with Parts Using PDM (WRKPARTPDM)*** panel, press **F16** (***User-defined options***). Then create a user-defined option with this command:

```
SOFTTURNE/TDSPBLDOPT FILE(&L/&F) MBR(&n)
```

c. On the ***Work with Parts Using PDM (WRKPARTPDM)*** panel, use **F15** to filter the panel to show only BLDOPT parts.

d. Select a BLDOPT part with your user-defined option to see which commands have been customized. The command helps you determine which customized creation commands you are using. You will want to review the commands within each QDFT BLDOPT part and within each BLDOPT part with a compiler name.

e. Use this information to customize the TURNOVER® for iSeries v100 global type code command. You should make sure you have assigned the appropriate variables to keywords that are important to your environment.

f. You will need to analyze creation commands that have been entered for specific objects. Normal TURNOVER® for iSeries v100 processing may already account for the customizations you have made to your ADM commands. For example, if you have set the USRPRF keyword to ***OWNER** in a customized creation command, such as the **CRTBNDRPG** command, you would not have to do this for TURNOVER® for iSeries v100. TURNOVER® for iSeries v100 will recognize that the ***PGM** object that is being replaced has already been defined with ***OWNER** and will recreate the program with this same configuration.

g. As a general rule, you need only be concerned about options that can be specified only at compile-time. TURNOVER® for iSeries v100 automatically handles anything that can be changed after the object has been compiled, such as with the **CHGPF** (**Change Physical File**) or **CHGPGM** (**Change Program**) commands. Training services from UNICOM Systems, Inc. can help you decide which command overrides may or may not be necessary once you have migrated to TURNOVER® for iSeries v100.

UNICOM Systems, Inc. Recommends

We also strongly recommend that, before you proceed, you thoroughly review your TURNOVER® for iSeries v100 application definitions and migration plan.

6) Export parts

Use TOADMUTIL menu option **11** (***Work with Parts for Export***) to see a list of parts in your projects. The parts list provides you with the following information:

- The TURNOVER® for iSeries v100 application that is mapped to each ADM part.
- The TURNOVER® for iSeries v100 type codes that are mapped to the parts.
- If there are any errors in the mappings you have created.

Using the mappings that are defined, you can determine if the parts exist in the libraries defined in your TURNOVER® for iSeries v100 application definition. If necessary, you can then add parts to an export job. The export job exports the objects and updates their object descriptions.

Export parts. The export parts process uses the ADM **EXPPART** command to export the specified parts from the ADM environment libraries to the libraries that are defined in your TURNOVER® for iSeries v100 application. It is possible that you might not need to export parts from some groups. Be sure to fully analyze, using the available fields on the ***Work with Parts for Export*** panels, which objects exist and where.

The export parts process fills in the primary values on the **EXPPART** command, based on the values entered. These are the **EXPPART** command fields that are filled in:

```
EXPPART PRJ(&Project) GRP(&Group) TYPE(&Type) PART(&Part)
LANG(&Language) TOLIB(&TargetLibrary)
SRCFILE(&TargetSourceFile)
```

You might need to modify this command with other command parameters, such as **DATA(*YES)** or **REPLACE(*NO)**. To do so, create a TEXPOVER(*CHAR 500) data area in your data library (SOFTTURND, or your custom library name) in which you list the additional parameters you want to append to the end of the **EXPPART** command shown above.

Update object descriptions. The update object descriptions process updates the object descriptions of the specified object, or of the object that relates to the specified source. It does this by examining the type code mapped to the part and determining which source code is related to which part. The process then does the following:

- Stamps the object description with the TURNOVER® for iSeries v100 type code and the source information (specifically, the source file, library, and member).
- Updates the object's source change date to match the actual change date of the corresponding source. This is necessary because the **EXPPART** command could export an object type and a source type, but the object's object description will still point to the source code that actually created it, regardless of the source's corresponding movement.

For each ADM part, you can choose to export the parts, update the object descriptions, or both.

In addition to (or instead of) change object description processing, you can define an exit program to do your own processing. With this exit point, you can control any additional processing your setup may require. Your exit program can pass back a flag to notify TURNOVER® for iSeries v100 if the process should or should not perform the standard change object description processing.

To have the exit point call your program, first create a TEXPJOBX(*CHAR 20) data area in your data library (SOFTTURND, or your custom library name). In the first 10 positions of the data area, supply your program name. In the second 10 positions, supply the name of the library where the program exists. Your program should then define these parameters:

Parameter	Type	Length
Object	*CHAR	10
Library	*CHAR	10
Object Type	*CHAR	10
Source File	*CHAR	10
Source Library	*CHAR	10
Source Member	*CHAR	10
Type Code	*CHAR	10
Flag	*CHAR	1

Pass back a value in the FLAG parameter to control normal processing, as follows:

Blank value	Continue normal processing (stamp the object description).
Non-blank value	Bypass normal processing (don't stamp the object description).

6.1) Steps for selecting parts for export

a. Use TOADMUTIL menu option **11** (***Work with Parts for Export***) to specify your ADM project to get a list of parts.

b. Use **F11** to show additional information. You can use the information listed to determine whether or not the part exists in the library defined by the mapping rules you have established. You may or may not have to export all parts.

c. Select, with option **11** (***Export***), option **12** (***Update object description***), or option **13** (***Both***), the parts that you will export. The selected records will be added to the export job defined in the ***Use export job*** field.

6.2) Steps for running the export job

a. Use TOADMUTIL menu option **12** to see the export jobs that are defined.

   If you have parts that defaulted with a type code that needs to be changed, you can do that here. For example, you might have an RPGLE type code which points to a library that applies to 99% of your programs, but not to the few programs needing to go into a different library. In this case, you would define another type code for this alternate promotion path and then assign that type code here. Also, if you need to, you can further customize the **EXPPART** command here by adding whatever additional **EXPPART** command parameters you need.

b. Once you verify that the parts selected have been set up the way you intend, you can select the export job to run. The **SBMJOB** command prompt appears so you can change the JOBQ, run date and time, or other parameters for the **SBMJOB** command, if necessary.

   Note: If for some reason, you need to cancel the export job, use option **4** (***Cancel***) on the ***Work with Export Jobs*** panel. This allows the export job to finish the line it is currently exporting or updating, and to come to a normal end.

7) Perform TURNOVER® for iSeries v100 initial build

The TURNOVER® for iSeries v100 initial build loads the TURNOVER® for iSeries v100 object history and audit database with initial entries for the application and level. TURNOVER® for iSeries v100 uses these records to detect unauthorized access to production objects and source.

To submit the initial build, use TURNOVER® for iSeries v100 Main Menu option **1** (***Work with application definitions***) and select an application with option **2** (***Change***). Select an application level with option **23** to submit an initial build for it.

UNICOM Systems, Inc. Recommends

We recommend that you perform the initial build for the highest unlocked level (Production) of the application. You should perform this step only once, after you have migrated all objects from your ADM environment for use with TURNOVER® for iSeries v100 and you have updated the object descriptions to point to the new source location.

8) Assign pre- and post-run commands

When you compile some objects, it is important that you run commands before or after the compile command. A typical example would be using the **OVRDBF** (**Override with Database File**) command to compile an object over a particular file, then using **DLTOVR** (**Delete Override**) to remove the override.

ADM holds these commands in a Build Option (BLDOPT) part. In TURNOVER® for iSeries v100, you enter pre-run or post-run commands on a form and link them to a particular object. When you run the form, TURNOVER® for iSeries v100 issues the commands.

To see the commands you have defined to ADM, use the **TDSPBLDOPT** (**Display Build Options**) command as a user-defined option on the ***Work with Parts Using PDM (WRKPARTPDM)*** panel. From **TDSPBLDOPT**, press **F16** to access the ***User-defined Options*** panel. Look for the user-defined option for **TADDCMDOBJ**:[2]

```
TADDCMDOBJ CMDTYPE(*PRERUN) LINE(*SELECT) CMD('&CMD')
SEQ(*NEXT)
```

To use this command, you must know the number of the form containing the object to which the selected command should be linked. TURNOVER® for iSeries v100 created these forms in the initial build step.

[2] If you do not find the user-defined option for **TADDCMDOBJ**, then you can add it using **F6**.

If you know the object name, target library, and TURNOVER® for iSeries v100 type code, or if you know the specific line number for the object, you can just supply that information to **TADDCMDOBJ** rather than use ***SELECT**. Keeping the default of **LINE(*SELECT)** shows you all lines on the selected form, so you can choose the object to which the command should be linked.

Note that some pre- or post-run commands may not be necessary once you start using TURNOVER® for iSeries v100. For example, when TURNOVER® for iSeries v100 promotes a file, it recreates the new file with the same members that existed in the old file. If you have previously used an **ADDPFM** (**Add Physical File Member**) command for a particular file, you would not have to this for TURNOVER® for iSeries v100. Our training services can help you decide which objects may or may not require the ADM commands specified.

Also note that if your processing uses the **OVRDBF** command, you should set the value of the **OVRSCOPE** parameter will need to make use of the OVRSCOPE parameter by setting the value to ***CALLLVL** or ***JOB**. This ensures that TURNOVER® for iSeries v100 properly scopes the override during the compile.

9) Review work in progress

Perform this step if you have checked out parts that you could not promote to the root group before you migrated to TURNOVER® for iSeries v100. For any in-process work, you should:

1) Create a task in the TURNOVER® for iSeries v100 project system that logically groups the work that is still in progress.

2) Assign each task you have created to the programmer who is working on the objects.

The programmer should then:

1) Create a TURNOVER® for iSeries v100 worklist for each task.

2) Add the objects that s/he is working on to the corresponding worklist by using a user-defined option in ADM.

3) Check out these objects using worklist option **21**. To place the source into his/her new development source file and library, depending on where the source exists for the objects, the programmer can either use the **CPYOPT** parameter or manually copy the source from the ADM development group.

SUMMARY

In this document, we have introduced you to TURNOVER® for iSeries v100, discussed ADM and TURNOVER® for iSeries v100 terminology and their correspondence, described the major differences in ADM's and TURNOVER® for iSeries v100's handling of change management, and explained the developments that have placed TURNOVER® for iSeries v100 in a leadership position for adapting change management to WebSphere application development. In the second half of the document, we provide detailed steps for migrating from the ADM to TURNOVER® for iSeries v100 change management systems.

Again, if you have any questions about this information, please do not hesitate to contact a UNICOM Systems, Inc. Technical Support Representative via phone, fax, or email at the locations shown at the beginning of this document.