

Supplement #14

USA: +1 603 924 8818 • techsupport@softlanding.com • <https://support.softlanding.com>

EMEA: +44 1293 872101 • softlandingsupport@macro4.com • <https://support.softlanding.com>

THE TURNOVER® FOR ISERIES CHANGE PROCESS

TABLE OF CONTENTS

INTRODUCTION	2
TURNOVER® FOR ISERIES FORM PROMOTION PROCESSING FLOW	4
Rolling back a form.....	8
Rerunning a failed form	8
SOURCE-ONLY OBJECTS.....	9
SOURCE-COMPILED OBJECTS	10
NON-SOURCE OBJECTS	11
ADDITIONAL RULES AND CONDITIONS.....	12
TURNOVER® for iSeries methods.....	12
CREATE parameter and initial object authority rules.....	13
Create parameters.....	13
Initial object authority	13
Logical files.....	14
Deleting a physical file with logicals.....	15
Multiple-membered physical files	15
Multiple-membered logical files.....	15
NEW LOGICAL FILES	16
Duplicating and moving logicals	16
Journaling files	16
Promoting programs	17
IBM i menus	17
S/36 menus and DFUs.....	17
Processing commands.....	17
PRE-RUN AND POST-RUN COMMANDS	18
Pre-run commands	18
Post-run commands.....	18
PRE-RUN AND POST-RUN COMMAND SCENARIOS	19
DELETING OBJECTS.....	19
FORM COPY AND OBJECT DELETE RULES	20
Background	20
Form copy rule.....	20
DELETE TEST OBJECT RULE.....	22

INTRODUCTION

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

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

1. Source-only “objects”
2. Source-compiled objects
3. Non-source objects
4. Non-IBM i types (such as CASE tool interface types).

Generally, all objects in each group are handled in the same way; but there are exceptions. Most notably, physical files with associated logical files, and message files.

The key variables that control TURNOVER® for iSeries’ change process are:

Method	(CSCO, CSMO, CSCD, CS, CD, MO, COPR, *NULL)
Action code	(A=Add, R=Replace, D=Delete)
TURNOVER Type code	(Found in the <i>TURNOVER® for iSeries</i> type code file of the application)
Object type attribute code	(RPG, CLP, PF, LF, and so on)
Create Command	(Found in the TurnOver global type code file)

While several other variables are used throughout the process, these are the most important.

Before continuing with this document, we suggest that you review the ***Working with Type Code Definitions*** section in ***Chapter 8: Utility Menu*** of the ***TURNOVER® for iSeries User Guide***.

The following page lists the steps performed by the form promotion job. A promotion job consists of four phases:

1. Pre-run processing
2. Promotion processing
3. Post-run processing
4. Finalization.

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

The TURNOVER® for iSeries Change Process

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

TURNOVER® FOR ISERIES FORM PROMOTION PROCESSING FLOW

TURNOVER® for iSeries form(s) selected to run

All forms are set to IN-USE. Batch job named TOnnnnnnnG is submitted to process form(s).

TURNOVER® for iSeries form error-checking (all forms)

[Note that most of these checks are performed when the form is submitted or during the error-checking process. This process is usually done in batch. If only one form is run, you may elect to do error-checking interactively.]

1. Check user's authority to submit a form.
2. Look in checkout file for missing objects. The form's **Project task reference** code is used to look for objects checked out with the same reference that are not included on the form.
3. Check to see if test member and object match. Object description date and times are used to ensure a match (where Method is CSCD or CSMO only).
4. Check to see if test members or objects have changed since form was approved.
5. Check to see if production member exists (must exist, if replace or delete; can exist if add and will be archived).
6. Check to see if production object exists (must exist, if replace or delete; must not exist if add).
7. Check program references. Will required files be found in run-time library list?
8. Check to see if source member had been checked out.
9. Check programmer.
10. Check object name against naming mask (new objects only).
11. Check to see if reference (REF*) objects exist.
12. Check to see if the from and target source files are the same length (S/36 environment only).
13. Check to see if the same object name, target library and type is on more than one form (except if first occurrence is a delete).
14. If any form has errors, TURNOVER® for iSeries produces an error log and returns all forms to **READY** status. No further processing occurs.
15. Call the Pre-Run Error Check Exit program (Exit 13).
16. If no errors exist, call the Pre-Form Group Exit program (Exit 19).
17. Run any TestBench ***PRERUN** tests before any objects are promoted. Form will remain in **IN-USE** status during this phase of processing.

TURNOVER® for iSeries Form Pre-run processing (Phase I)

1. Set form status to **RUNNING**.
2. Check form dependency. If dependent form failed or has not been run, terminate job.
3. Check date-time range specified in form header.
4. Check for object locks and allocate, if specified.
5. Call interface programs, if appropriate.
6. Call the Form Initialization Phase Exit program (Exit 8), if specified.

The TURNOVER® for iSeries Change Process

7. Run pre-run commands specified as ***PREFORM**. Overrides associated with one form will not be in effect for other forms.

TURNOVER® for iSeries Promotion processing (Phase II)

1. Promote IFS objects with a ***BEFORE** sequence in the global type code.
2. Create archive library TnnnnnnDy, where nnnnnn=form number and y=ASP. (Multiple libraries may be created if applications use multiple ASPs).
3. End physical and logical (access path) file journaling.
4. Remove pre-existing triggers and/or constraints.
5. Copy source and move objects to TnnnnnnD library(s). (If program type object and creation command parameter **REPLACE** is ***YES**, TURNOVER® for iSeries will create duplicate of production object into TnnnnnnD archive library).
6. Copy source to target source files and libraries. If method is COPR, source is not copied to target library.
7. Run pre-run commands associated with the line if specified as ***PRELINE**, using library list ***CURRENT** or ***APPL**, depending on form command definition.
8. Replace library list with application library list.
9. One of the following:
 - Compile object into target library using target source.
 - Move test object to target library.
 - Duplicate test object into target library.
10. Run post-run commands associated with the line if specified as ***POSTLINE**, using library list ***CURRENT** or ***APPL**, depending on form command definition.
11. Change creation parameters.
12. If physical or logical file, add members.
13. If physical files, copy data from old file into each target file member.
14. Promote IFS objects with an ***AFTER** sequence in the global type code.

No Errors detected: If none of the pre-run or mainline steps cause the form to end, then the next form will be set to **RUNNING** and these steps will be repeated on that form. Status of the current form will remain **RUNNING** after these steps have completed.

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

The TURNOVER® for iSeries Change Process

TURNOVER® for iSeries Post-run processing (Phase III)

(All Phase II is completed for all objects and forms before Phase III begins.)

1. Set owner of new object.
2. Set authority of new object.
3. Restart file journaling, if appropriate.
4. Replace pre-existing triggers and/or constraints.
5. Explode data objects (repeated for all data type objects in each explode library).
 - Create data explode archive library, TnnnnnnE (once).
 - Move data object into archive library. Create alias name, if duplicate.
 - Duplicate physical files from primary target library into explode library.
 - Create members of physical files in explode library and copy data.
 - Create logical files in explode library.
 - Set Create parameters of logical file(s).
 - Add members to logical files.
 - Set authority and ownership of explode library objects.
6. Call interface post-run programs.
7. Call the Form Finalization Processing Exit program (Exit 1), if appropriate.
8. Call post-run commands specified as ***POSTFORM**.
9. Set form status as follows:
 - **FIN-WRN** = Finalization phase, warning messages generated during processing.
 - **FIN-ERR** = Finalization phase, error messages generated during processing (one or more objects not promoted).
 - **FIN-OK** = Finalization phase, no errors detected.
10. If auto-distribution and status is **FIN-OK** for all forms in the group, submit distribution job.
 - Status will be set to **FIN-PND** (distribution pending) if finalization has not yet completed, or to **DIST-PND** if finalization has completed.
 - After distribution completes, the status will be set to **FIN-DIST**, **FIN-PART**, **DISTRIB**, or **PARTIAL**.
 - When confirmation is received, the status will be set to **FIN-CONF** or **CONFIRM**. Any **FIN-** status will be updated at the end of the finalization job. If distribution is automatic, the forms will be distributed as a form group. If the remote computer is set up to run forms automatically, they will be run as a form group when they are received.

The TURNOVER® for iSeries Change Process

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

TURNOVER® for iSeries Finalization processing (Steps 1-6 are done in a separate finalization job if *Submit finalization* is set to **Y**)

1. Archive source.
2. Delete test source member.
3. Delete test object.
4. Rebuild cross-referencing.
5. Delete TnnnnnnnD libraries and TnnnnnnnE libraries, if *Archive Objects* is **N** and no errors were detected.
6. Set form status to **RAN-OK**, **RAN-WRN**, or **RAN-ERR**.
7. Send completion messages to host computer (if a distributed form).
8. Send form completion messages.
9. Call the Form Completion Processing Exit program (Exit 11).
10. Run any TestBench ***POSTRUN** tests that have been specified.
11. Call the Post Form Group Exit program (Exit 20).
12. Print TURNOVER® for iSeries logs.

Note that most of these steps are conditional and depend on the four major control variables mentioned earlier.

The diagrams on the following pages illustrate the processing steps for the each of the three generic object types: source-only, source-compiled objects and non-source objects. Because of the variability of non-IBM i types, these are not illustrated here. Refer to the respective TURNOVER® for iSeries Interface User Guide for the interface product in question for specifics about promoted non-IBM i object types.

Rolling back a form

If a TURNOVER® for iSeries form fails and you specified that TURNOVER® for iSeries should roll back the changes, TURNOVER® for iSeries reverses the change process, copying and moving source and object back from the target libraries to the from libraries and from the archive library(s) to the target libraries.

You can also roll back a form after it has completed successfully by creating and running a recovery form. This procedure requires that objects are being archived and that they are on-line at the time you run the recovery form. When you create a recovery form, you can specify that it contain only items that were promoted; this saves you from having to edit and remove other items after you create the form. Also, if any physical files on the form have been updated since the form ran, be aware that the data will be copied from the new file to the old file using the copy command parameter **CPYFMT** of ***MAP *DROP**. See the *Creating a Recovery Form* section in *Chapter 7: Working with TURNOVER Forms* in the *TURNOVER® for iSeries User Guide*.

Rerunning a failed form

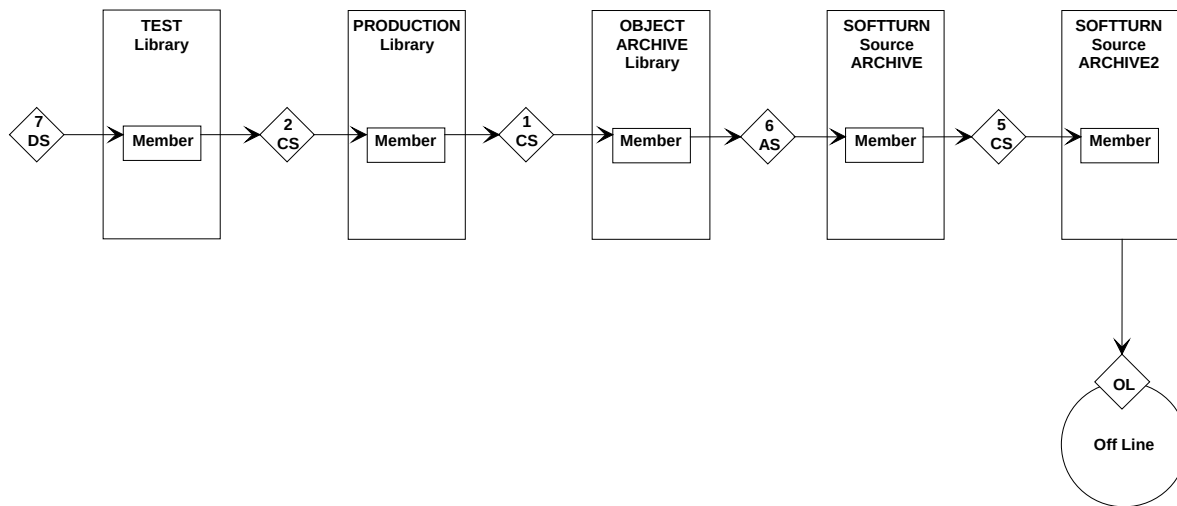
If a form fails because someone canceled the job, your power or computer failed while the form was running, or for any other reason where the form status is **RUNNING**, even if the job is not running, you can restart the form. You do so by selecting the form with option **8=Work w/ job** on the *Work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

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

UNICOM Systems, Inc. Caution!

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

The TURNOVER® for iSeries Change Process



SOURCE-ONLY OBJECTS

By far the easiest to describe, source-only objects such as help text and copybook code, can be promoted to production using only one TURNOVER® for iSeries **Method, CS (Copy Source)**. (Steps occur in the sequence indicated.)

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

Legend

CS=Copy Source

MO=Move Object

CD=Create Duplicate Object

CO=Compile or Create Object

AS=Compress and Archive Source

DS=Delete Source

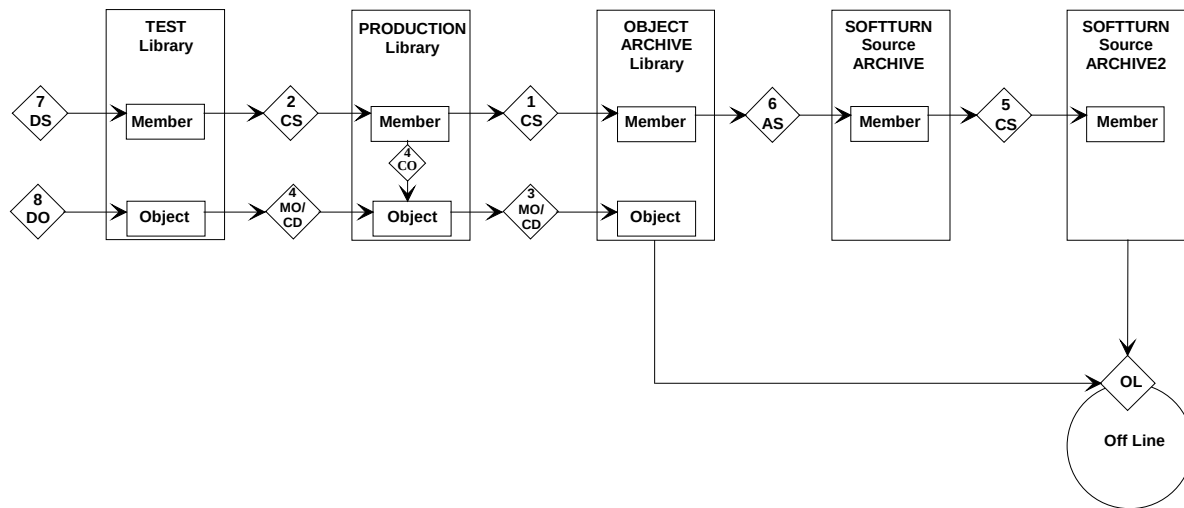
DO=Delete Object

OL=Off-Line Save

SOURCE-COMPILED OBJECTS

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

If you use **CSMO** or **CSCD**, TURNOVER® for iSeries will make sure that the source and object match before allowing you to submit the form. It also checks to see if the source or object has changed since it was submitted, to prevent tampering with or substitution of another source member or object after job submission. (Steps occur in the sequence indicated.)



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

Legend

CS=Copy Source

MO=Move Object

CD=Create Duplicate Object

CO=Compile or Create Object

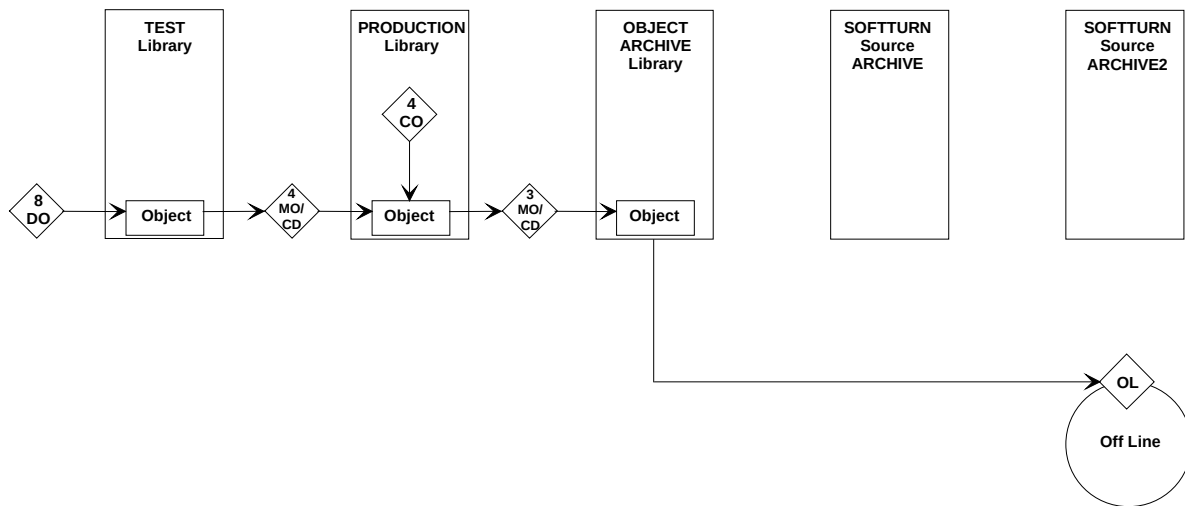
AS=Compress and Archive Source

DS=Delete Source

DO=Delete Object

OL=Off-Line Save

The TURNOVER® for iSeries Change Process



NON-SOURCE OBJECTS

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

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

Legend

CS=Copy Source

MO=Move Object

CD=Create Duplicate Object

CO=Compile or Create Object

AS=Compress and Archive Source

DS=Delete Source

DO=Delete Object

OL=Off-Line Save

ADDITIONAL RULES AND CONDITIONS

The remainder of this document describes additional rules, conditions, and suggestions to consider when you are working with TURNOVER® for iSeries.

TURNOVER® for iSeries methods

The **method** you specify in your application line defaults controls how TURNOVER® for iSeries transfers source or objects (or both) from test to production. Your choices are:

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

CREATE parameter and initial object authority rules

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

Create parameters

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

Initial object authority

See the TURNOVER® for iSeries Supplement entitled *Security Recommendations* (#15).

Logical files

You do not have to explicitly delete a logical file over a physical file before recreating or deleting the physical file. TURNOVER® for iSeries will delete logical files automatically. When you enter a physical file on a TURNOVER® for iSeries form, TURNOVER® for iSeries will insert a line on the form for each logical over the physical being deleted or replaced. This is done after you have entered all the lines on the TURNOVER® for iSeries form. When you exit from the **Form Maintenance** panel, you'll see the message "**Checking data base relations for PF on form**". TURNOVER® for iSeries verifies and adds entries for logical files at this time. Logical files already on the form will not be duplicated.

How the logical file logic works

The logical file logic works like this:

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

Does a logical file exist?

Yes: Is there a delete for this logical file on this form?

Yes: Do nothing.

No: Is the logical file created earlier on this form?

Yes: Move the line to create the logical file to the line following the create physical file line.

No: TURNOVER® for iSeries inserts a line to create the logical file.

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

Deleting a logical file

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

Deleting a physical file with logicals

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

Multiple-membered physical files

TURNOVER® for iSeries will preserve all members of an existing physical file. You're responsible for adding new members or removing obsolete members. You can do so by putting a post-run command on the form.

Existing physical files

If a multiple-membered file is being replaced, TURNOVER® for iSeries will recreate the file with all the members that were in the physical file. All of the members will be preserved and copied in the new version of the file using the copy command with the format options ***MAP** and ***DROP**.

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

New physical files

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

If data needs to be copied into a member, you can call a program, use a copy command or use the **CPYTSTDTA** command. For more information, see the TURNOVER® for iSeries Supplement entitled *TURNOVER® for iSeries Commands (#28)*.

Multiple-membered logical files

TURNOVER® for iSeries will preserve all members of an existing logical file. You're responsible for adding or removing members.

Existing logical files

If the multiple-membered logical file is being replaced, TURNOVER® for iSeries will recreate the file with all the members that were in the original logical file.

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

NEW LOGICAL FILES

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

Duplicating and moving logicals

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

UNICOM Systems, Inc. CAUTION!

We recommend that you exercise extreme caution when overriding the recreate rule.

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

Journaling files

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

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

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

You are responsible for starting journaling for any file or access path being added. You can do this with a post-run command. If you are journaling files in a series of explode libraries, use the “**&ALLTGT**” (including the quotes) parameter. For more information, see **Pre-Run and Post-Run Commands** on page 18.

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

Promoting programs

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

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

IBM i menus

IBM i menu object types are supported by TURNOVER® for iSeries. During checkout, TURNOVER® for iSeries determines the type of menu (message file or program) and copies the message file and other menu components or source to the development library. When you build the TURNOVER® for iSeries form using the reference-build feature, TURNOVER® for iSeries includes lines for each menu component on the form. For more information, see the TURNOVER® for iSeries Supplement entitled *Menus, DFUs, and Queries* (#58).

S/36 menus and DFUs

TURNOVER® for iSeries provides a special checkout option for S/36 menus and DFUs. When you create a form using the reference build feature, TURNOVER® for iSeries includes lines for each menu or DFU component on the form. For more information, see the TURNOVER® for iSeries Supplement entitled *Menus, DFUs, and Queries* (#58).

Processing commands

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

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

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

PRE-RUN AND POST-RUN COMMANDS

Pre-run commands are run at the beginning of a TURNOVER® for iSeries form job (***PREFORM**) or just before an object on the form, associated with the command, is promoted (***PRELINE**). Post-run commands are run after all objects are promoted (POSTFORM), or just after an object associated with the command is promoted (POSTLINE). Exactly when the command is run depends on the parameter you set when you enter the pre/post-run command on the TURNOVER® for iSeries form.

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

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

UNICOM Systems, Inc. Note

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

Pre-run commands

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

Post-run commands

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

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

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

PRE-RUN AND POST-RUN COMMAND SCENARIOS

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

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

You might want to do a **SAVLIB** or **SAVOBJ** before starting a TURNOVER® for iSeries. This can be entered as a pre-run command.

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

You might want to copy data from a test file to production. This can be done using the **CPYTSTDTA** command as a post-run command. For more information, see the TURNOVER® for iSeries Supplement entitled ***TURNOVER® for iSeries Commands (#28)***.

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

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

For information about including pre- and post-run commands on a TURNOVER® for iSeries form, see ***Chapter 6: Working with TURNOVER® for iSeries Forms*** in the ***TURNOVER® for iSeries User Guide***.

DELETING OBJECTS

To delete an object from production, add the item to the TURNOVER® for iSeries form set the Action code to **D (Delete)**.

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

TURNOVER® for iSeries will move the object to the ***Archive library*** and will archive any existing source, before it deletes the object and source from the ***Target library***.

FORM COPY AND OBJECT DELETE RULES

This section defines the rules TURNOVER® for iSeries follows when:

- Copying forms from one application level to a higher application level.
- Determining whether or not to delete test *From objects* when the TURNOVER® for iSeries form is run.

Background

TURNOVER® for iSeries adds lines to forms automatically during a cross-reference check (if you use a cross-reference facility) or by checking database relationships for physical files. TURNOVER® for iSeries adds a line to the form for each related object it finds, and sets the *method* for these lines to **COPR** – which compiles the object into the target library for this level of the application, using the source available either in the target source library or at a higher level. For example, if you are promoting a file to an intermediate test level, TURNOVER® for iSeries looks for programs created over that file at that test level and at the higher production level. If a related object is found, it is added to the form as method **COPR** to be created in the test library from the source in the production library.

What happens when this form is copied to the next higher level?

Form copy rule

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

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

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

TURNOVER® for iSeries issues this message when it drops an object during the copy process:

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

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

The TURNOVER® for iSeries Change Process

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

DELETE TEST OBJECT RULE

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

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

If Global parameter *Use delete test object rule* is **N**, no further action required.

Else

If the *From object* is checked out to another *Programmer*

or the *From object* is checked out under a different *Project reference*,

set *Delete test object* to **N**.

Else

If *From object type* is a program

and a cross-reference exists for the From-application level

then, for each file used by the program, do the following:

If the file exists in the from-library

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

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

Then set *Delete test object* to **N**.

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

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

If during the form run TURNOVER® for iSeries found it necessary to change *Delete test object* from **Y** to **N**, you will find this message in the joblog:

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

The TURNOVER® for iSeries Change Process

The reason codes for this message also appear in the joblog, and coincide with the conditions described above. They are:

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

If you have any questions about the information in this document, please contact a UNICOM Systems, Inc. Technical Support Representative by phone, fax, or email at the locations shown at the beginning of this document.