SoftLanding SYSTEMS
A Division of the UNICOM Group of Companies

UNICOM® UNICOM SYSTEMS, INC.

Supplement #59

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

MANAGING USER-DEFINED OBJECT TYPES

TABLE OF CONTENTS

This document describes how you can manage user-defined object types to selectively override change management methods that TURNOVER® for iSeries v100 executes for an object.

OVERVIEW

UNICOM Systems, Inc. Caution

The information in this document is intended for advanced TURNOVER® for iSeries v100 users. If you are not thoroughly familiar with TURNOVER® for iSeries v100, you should not work with change management schemes without first discussing your goals with a UNICOM Systems, Inc. Technical Support Representative, Trainer or agent. Chances are very good that the base TURNOVER® for iSeries v100 features already support what you want to accomplish.

Since its inception, TURNOVER® has performed a significant amount of processing steps in order to correctly and safely promote the source and object for any given type code. As a matter of fact, there are sixteen individual steps, or “methods” that TURNOVER® for iSeries v100 uses during. For example,

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

These sixteen methods are collectively referred to as the TURNOVER® for iSeries v100 Change Management “scheme” for that type code. TURNOVER® for iSeries v100 always executed these methods using default system commands such as **MOVOBJ**, **CRTDUPOBJ**, **CHKOBJ**, and so forth.[1]

With TURNOVER® for iSeries v100, you can create your own *change management schemes* to define any of the sixteen *change management methods* that you would execute for an object type. For example, you may want to write a CM Scheme that will manage the promotion of certain data records in one of your company’s key database files. Or a CM scheme that would enable some enhanced promotion processing for SQL. Or a CM scheme that would use IBM’s CHGPF data conversion method instead of the more traditional methods such as *MAP *DROP.

Several sample change management schemes are shipped with TURNOVER® for iSeries v100. You can use these schemes as examples or create your own schemes. These schemes include sample source programs that you can browse to better understand the concepts. The sample schemes are described beginning on page 28.

[1] If the object type had a record in the TURNOVER® for iSeries v100 User-defined Type file (TUSRTYPF), TURNOVER® for iSeries v100 bypassed these methods. For more information about the TUSRTYPF file and its relationship to CM schemes, see ***Relationship between the TUSRTYPF file and CM schemes*** on page 26.)

WHAT YOU NEED TO DO

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

1) Create a change management scheme. (See ***Step 1: Create a CM Scheme*** on page 5.)

2) Override change management methods for the new change management scheme (as necessary). (See ***Step 2: Override CM Methods*** on page 8.)

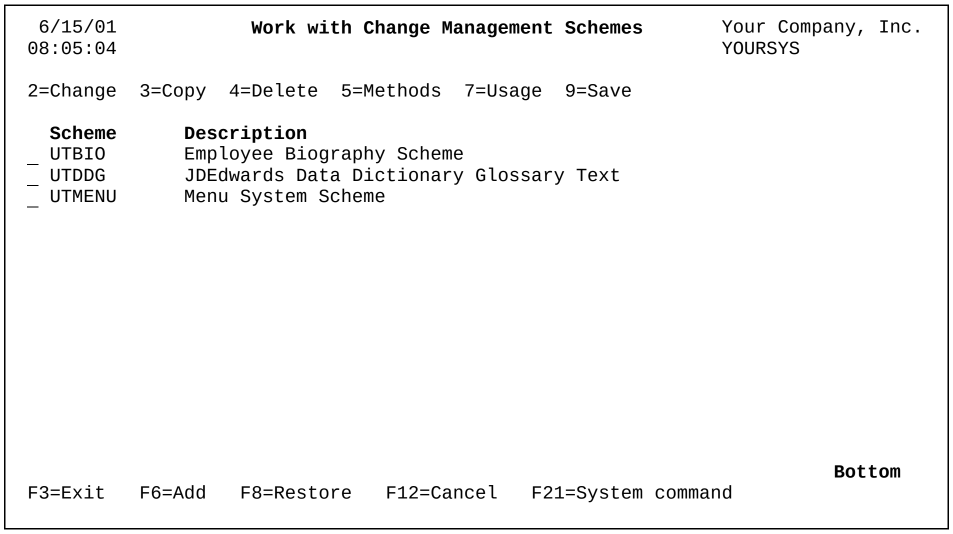
3) Create *change management method programs* for change management methods you have overridden. (See ***Step 3: Create CM Method Programs*** on page 17.)

4) Create new TURNOVER® for iSeries v100 type codes or change existing TURNOVER® for iSeries v100 type codes to use the new change management scheme. (See ***Step 4: Create Type Codes to Use a CM Scheme*** on page 28.)

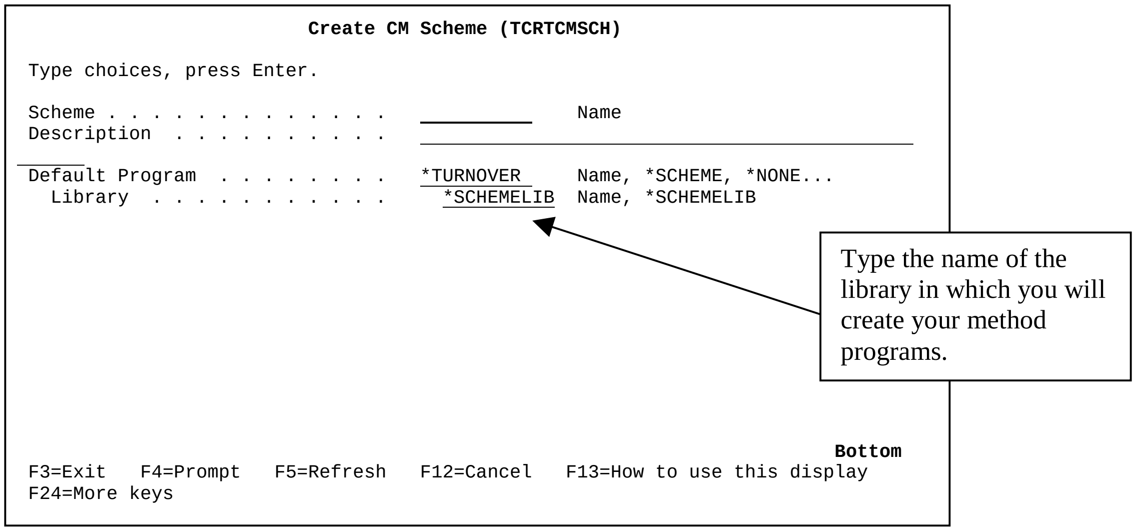
These steps are described in detail in this document.

STEP 1: CREATE A CM SCHEME

To work with change management (CM) schemes, type **TWRKCMSCH** on a TURNOVER® for iSeries v100 command line and press **Enter**. The ***Work with Change Management Schemes*** panel appears:


```
 6/15/01            Work with Change Management Schemes       Your Company, Inc.
08:05:04                                                      YOURSYS

2=Change  3=Copy  4=Delete  5=Methods  7=Usage  9=Save

  Scheme      Description
_ UTBIO       Employee Biography Scheme
_ UTDDG       JDEdwards Data Dictionary Glossary Text
_ UTMENU      Menu System Scheme






                                                                        Bottom
F3=Exit   F6=Add   F8=Restore   F12=Cancel   F21=System command
```

This panel lists all the CM schemes that have been defined to TURNOVER® for iSeries v100. (Details about other options on this panel are described beginning on page 29.) Press **F6=Add** to create a new scheme. You must have TURNOVER® for iSeries v100 **System Defaults** authority to create CM schemes. The ***Create CM Scheme (TCRTCMSCH)*** panel appears:


```
                           Create CM Scheme (TCRTCMSCH)

Type choices, press Enter.

Scheme . . . . . . . . . . . . .   __________    Name
Description  . . . . . . . . . .   ____________________________________________

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






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

Type the name of the library in which you will create your method programs.

On the ***Create CM Scheme*** panel, supply the information TURNOVER® for iSeries v100 needs to create a CM scheme. Descriptions of the fields follow:

Scheme

Supply the name you want to give the CM scheme. Once you create a CM scheme, you can update TURNOVER® for iSeries v100 global type code definitions to refer to it by this name.

Description

Supply a description of the CM scheme.

Default Program

Each scheme is comprised of sixteen individual methods. For ease of implementation, all methods are originally set to invoke the *DEFUALT program in the *DEFUALT library. These two parmameters set the value of this scheme's default method processing program. (For information about method programs and libraries, see ***Changing a CM method*** on page 9.) Your choices are:

Name	Type a specific "hard coded" default method program name.
***SCHEME**	The default method program is the same name as the scheme.
***NONE**	For any method set to *DEFAULT, TURNOVER® for iSeries v100 will perform *no processing whatsoever*. For example, if you are writing a scheme that handles handles objects with no source code, you might want to instruct TURNOVER® for iSeries v100 to do no processing on the Archive Source method.
***TURNOVER**	Use TURNOVER® for iSeries v100's standard processing methodology. This is useful if you need to customize the processing for only a few of the methods and all of the others you'll defer back to TURNOVER® for iSeries v100's control. (For more information, see page 8.)

Default Program Library

Your choices are:

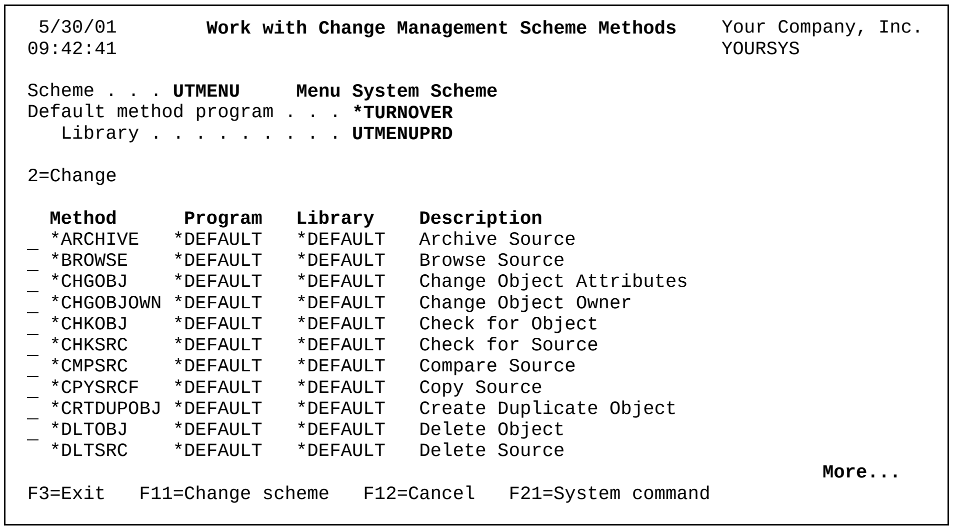
Name	Type a specific "hard coded" default method library name.
***SCHEMELIB**	TURNOVER® for iSeries v100 will retrieve the library name from the TSCHEMELIB data area (which is originally set to ?????).

When you finish supplying the scheme information, press **Enter**. TURNOVER® for iSeries v100 creates the scheme and adds it to the list on the ***Work with Change Management Schemes*** panel.

STEP 2: OVERRIDE CM METHODS

To fully customize your new CM scheme, you must override some of its CM methods to call your own processing program(s). To override a CM method, you need to change the name of the method program, the name of its library, or the names of both the program and library you want TURNOVER® for iSeries v100 to call when executing the method.

To begin working with your scheme's CM methods, select the scheme with option **5** on the ***Work with Change Management Schemes*** panel. The ***Work with Change Management Scheme Methods*** panel appears:



```
 5/30/01        Work with Change Management Scheme Methods    Your Company, Inc.
09:42:41                                                      YOURSYS

Scheme . . . UTMENU     Menu System Scheme
Default method program . . . *TURNOVER
   Library . . . . . . . . . UTMENUPRD

2=Change

  Method      Program   Library    Description
_ *ARCHIVE   *DEFAULT   *DEFAULT   Archive Source
_ *BROWSE    *DEFAULT   *DEFAULT   Browse Source
_ *CHGOBJ    *DEFAULT   *DEFAULT   Change Object Attributes
_ *CHGOBJOWN *DEFAULT   *DEFAULT   Change Object Owner
_ *CHKOBJ    *DEFAULT   *DEFAULT   Check for Object
_ *CHKSRC    *DEFAULT   *DEFAULT   Check for Source
_ *CMPSRC    *DEFAULT   *DEFAULT   Compare Source
_ *CPYSRCF   *DEFAULT   *DEFAULT   Copy Source
_ *CRTDUPOBJ *DEFAULT   *DEFAULT   Create Duplicate Object
_ *DLTOBJ    *DEFAULT   *DEFAULT   Delete Object
_ *DLTSRC    *DEFAULT   *DEFAULT   Delete Source
                                                                       More...
F3=Exit   F11=Change scheme   F12=Cancel   F21=System command
```

This panel lists all the CM methods for the scheme you selected.

At the top of the screen is the current setting for this scheme's default method processing program and library as specified/controlled in Step 1 above. Any method that lists *DEFAULT for its processing program and/or library name will use these values. Pressing F11 will take you back to these settings if you'd like to change them here.

Next, you see each of the sixteen methods that can apply to a promotion situation. For each method, you see the name of the method, the name of the program TURNOVER® for iSeries v100 calls to process the method, the name of the library where the method resides, and a description of the method.

For ease of use, all methods are initially set to *DEFAULT. This enables you to focus more on the "exceptions" than the "rule". The next section, ***Changing a CM method*** , describes how you do this.

UNICOM Systems, Inc. Note

When you distribute, using the Remote Configuration Wizard or a form, application definitions with global type code definitions that reference change management schemes, TURNOVER® for iSeries v100 does NOT automatically send the change management scheme definitions and method programs. To send change management schemes and method programs to remote computers, you must use the **TSAVCMSCH** (**Save CM Scheme**) and **TRSTCMSCH** (**Restore CM Scheme**) commands. For more information, see ***Saving a CM scheme*** on page 30 and ***Restoring a CM scheme*** on page 32.

Changing a CM method

Method descriptions

The nature of the object type represented by the CM scheme determines if you need to override a CM method. Each CM method is listed in this section, with a brief guideline for when you might need to override it. (For more detailed information, see the **Notes** following these descriptions. Also, see ***Step 3: Create CM Method Programs*** on page 17 and the sample CM schemes beginning on page 28.)

Method Name[2]	Description
***ARCHIVE** **(Archive Object's Source)**	TURNOVER® for iSeries v100's default handing for this method is to copy the source member to the source file that is specified in the application definition and to compress the source member there. You must provide a method program (or override this method to ***NONE**) if the object type managed by this scheme is an object type that has source associated with it (that is, the global type code creation command references the **"&SM"**, **"&SF"**, and **"&SL"** substitution variables) but the "source" for the object type is not stored in the form of a physical file (source file or database file) member. (For example, the "source" might be keyed database records in several files that are processed by a "compiler" to generate some kind of object.) **Note:** The description (immediately above) about providing a method program also applies to other methods relating to source: ***CHKSRC**, ***CMPSRC**, ***CPYSRC**, ***CPYSRCF**, and ***DLTSRC**. You can also override the ***CMPSRC** method to ***NONE** if the condition described above exists. **Note:** You do not have to provide a method program or override the ***ARCHIVE** method to ***NONE** if all applications that manage the object type handled by this scheme have ***NONE** for the ***Source archive file*** application default. This is one of several CM methods (the others are ***MISSING** and ***RTVOBJD**) that require that data (other than that returned by the ***Return Error Flag***) be returned to the calling program. (For more information, see ****ARCHIVE (Archive Object's Source) CM method*** on page 24.)
***BROWSE** **(Browse Object's Source)**	TURNOVER® for iSeries v100 uses command **STRSEU OPTION(5)** as the default method handler for this method. You can provide a method program to browse objects of this object type if the source or the object itself (if the object type is non-source) for the object type managed by this scheme is not stored in a form that can be edited using **STRSEU** (source physical file).

[2] The method name is the first parameter for all CM method programs (see ***Step 3: Create CM Method Programs*** on page 17).

Method Name	Description
***CHGOBJ (Change Object Attributes)**	TURNOVER® for iSeries v100 is hard-coded to change object attributes (based on production object, test object, reference object) based on object type. Before Release 5.3 introduced CM schemes, this functionality was only supported for objects of type ***PGM**, ***CMD**, ***MODULE**, ***SRVPGM**, and ***FILE**, with appropriate attributes being changed based on the object type. You must provide a method program if you use the scheme to manage an object type other than those mentioned above, and if it is necessary to make some kind of change to the objects attributes upon promotion. If the object type is one of the types mentioned above, or if no changes need to be made to the object upon promotion, you can set this method to ***TURNOVER**.
***CHGOBJOWN (Change Object Owner)**	TURNOVER® for iSeries v100 uses the **CHGOBJOWN** command as the default method handler for this method. You must provide a method program for this method if the object type managed by this scheme is not a valid value for the **CHGOBJOWN OBJTYPE** parameter. Alternatively, you can override this method to ***NONE** if you do not need to change object ownership for the object type.
***CHKOBJ (Check for Object)**	TURNOVER® for iSeries v100 uses the **CHKOBJ** command as the default method handler for this method. You must override this method if the object type (global type code definition) represented by the scheme is not a valid value for the **CHKOBJ OBJTYPE** parameter. For example, if the object type is ***MYOBJTYP**, you must override this method. If the object type is ***PGM**, you do not need to override this method.
***CHKSRC (Check for Object's Source)**	TURNOVER® for iSeries v100 uses command **CHKOBJ** (with **OBJTYPE** = ***FILE** and **MBR** = source member) as the default method handler for this method. (For more information, see the description of ***ARCHIVE** on page 10.)
***CMPSRC (Compare Source Members for Object Type)**	TURNOVER® for iSeries v100 uses the command found in the TOCMPSRC data area (shipped as **CMPSRCMBR**) as the default method handler for this method. (For more information, see the description of ***ARCHIVE** on page 10.)
***CPYSRCF (Copy Object's Source)**	TURNOVER® for iSeries v100 uses the **Copy File** (**CPYF**) command as the default method handler for this method. (This method includes checkout, promotion, copying archived source, and so forth.) (For more information, see the description of ***ARCHIVE** on page 10. Also, see ***CM method program considerations*** on page 26 for additional important information about method programs

Method Name	Description
	for this method.)

Method Name	Description
***CRTDUPOBJ (Create Duplicate Object)**	TURNOVER® for iSeries v100 uses the **CRTDUPOBJ** command as the default method handler for this method. You must override this method if the object type (global type code definition) represented by the scheme is not a valid value for the **CRTDUPOBJ OBJTYPE** parameter. Even if you have no intention of using TURNOVER® for iSeries v100 promotion methods **CD** (Create Duplicate Object) or **CSCD** (Copy Source, Create Duplicate Object) with object types managed by the scheme, you must still provide a method program if the object type cannot be duplicated using the **CRTDUPOBJ** command. This is because TURNOVER® for iSeries v100 uses **CRTDUPOBJ** to duplicate objects to a distribution work library when forms are distributed to other computers. (See ***CM method program considerations*** on page 26 for additional important information about method programs for this method.)
***DLTOBJ (Delete Object)**	TURNOVER® for iSeries v100 uses different commands to delete objects, depending on the object type. (The command to use for each object type is found in file TDLTOBJF.) Basically, if the object type is a valid parameter for the **CHKOBJ OBJTYPE** parameter, it is probably unnecessary to provide a method program for this method. The type is probably already handled by TURNOVER® for iSeries v100's default handling.
***DLTSRC (Delete Object's Source)**	TURNOVER® for iSeries v100 uses the **RMVM** (**Remove Member**) command as the default method handler for this method. (For more information, see the description of ***ARCHIVE** on page 10.)
***EDTSRC (Edit Object's Source)**	TURNOVER® for iSeries v100 uses command **STRSEU OPTION(2)** as the default method handler for this method, except when the Programmer Worklist Editor defaults are set to ***CODE400**. While it is not required, you can provide a method program to edit this object type if the source or the object itself (if the object type is non-source) for the object type managed by this scheme is not stored in a form that can be edited using **STRSEU** (source physical file).
***GRTOBJAUT (Grant Object Authorities)**	TURNOVER® for iSeries v100 uses a combination of several commands (**RVKOBJAUT**, **GRTOBJAUT**, and **CHGOBJPGP**) as the default handler for this method. This method includes setting object authorities upon promotion, based on an existing object, reference object, and so forth.

	You must provide a method program for this method if the object type managed by this scheme is not a valid value for the **GRTOBJAUT OBJTYPE** parameter. Alternatively, you can override this method to ***NONE** if it is not necessary to set authorities for the object type.
Method Name	**Description**
***MISSING (Create Missing Objects for Object Type)**	This is one of several CM methods (the others are ***ARCHIVE** and ***RTVOBJD**) that require that data (other than that returned by the ***Return Error Flag***) be returned to the calling program. (See ****MISSING (Create Missing Objects) CM method*** on page 23 for a description of the return data format for this method and for a description of how and when to override this method.)
***MOVOBJ (Move Object)**	TURNOVER® for iSeries v100's default method handler for this method is the **MOVOBJ** command. You must override this method if the object type (global type code definition) represented by the scheme is not a valid value for the **MOVOBJ OBJTYPE** parameter. Even if you have no intention of using TURNOVER® for iSeries v100 promotion methods **MO** (Move Object) or **CSMO** (Copy Source, Move Object) with object types managed by the scheme, you must still provide a method program if the object type cannot be moved using the **MOVOBJ** command. This is because TURNOVER® for iSeries v100 uses **MOVOBJ** to archive objects to the TnnnnnnnD library and to move them back from that library if TURNOVER® for iSeries v100 recovers a form or runs a recovery form. (On a remote computer, TURNOVER® for iSeries v100 also uses **MOVOBJ** to move objects to the TnnnnnnnS from-library when forms are received). (See ***CM method program considerations*** on page 26 for additional important information about method programs for this method.)
***RTVOBJD (Retrieve Object Description)**	TURNOVER® for iSeries v100 uses the **RTVOBJD** command as the default method handler for this method. You might want to provide a method program for this method (but it is not required) if the object type represented by the scheme is not a valid value for the **RTVOBJD OBJTYPE** parameter. For example, you might provide a method program to retrieve the text description of objects so the text will appear on the Programmer Worklist panel that displays the object description, and on several other TURNOVER® for iSeries v100 panels that display object descriptions. You should override this method to ***NONE** if the object type is not valid for **RTVOBJD** and a method program is not provided. (TURNOVER® for iSeries v100 does not attempt then to retrieve object descriptions for this scheme.) This is one of several CM methods (the others are ***ARCHIVE** and

	MISSING**) that require that data (other than that returned by the ***Return Error Flag) be returned to the calling program. (See ****RTVOBJD (Retrieve Object Description) CM method*** on page 21 for information about the proper format for returning object information from this method.)

Notes: Method programs that handle these methods must return a **Y** as its ***Return Error Flag*** parameter if it cannot successfully execute the method (for example, if for ***ARCHIVE** it cannot archive source or for ***CHGOBJ** it cannot change the object). (For information about the ***Return Error Flag***, see ***Step 3: Create CM Method Programs*** on page 17 and ***Error handling and messaging*** on page 21.)

Also, we recommend that you have the method program send informational messages to the calling program that describe exactly why the action was unsuccessful (for example, you could have a ***CHKSRC** method program return a message like ***"Definition for my special object MYOBJ not found in library MYDEVLIB"*** if the object's source does not exist. (See ***Error handling and messaging*** on page 21.)

If the method program does not successfully execute methods ***CHGOBJ**, ***CHGOBJOWN**, and ***GRTOBJAUT**, the TURNOVER® for iSeries v100 form runs with warnings.

Where the methods are used in TURNOVER® for iSeries v100

The following table summarizes where each CM method is used in TURNOVER® for iSeries v100.

Method Name	In TURNOVER® for iSeries v100, this method is used for …
***ARCHIVE** **(Archive Object's Source)**	❑ Archiving a source member during a form promotion. **Note:** Only applies if the global type code involves source (that is, the global type code creation command references the **"&SM"**, **"&SF"**, and **"&SL"** substitution variables).
***BROWSE** **(Browse Object's Source)**	❑ Browsing an object description by pressing **F2** on the ***Display Object Description (TDSPOBJD)*** panel. (**Note:** The **F2** key is available if the object is an object type that has source, and the source can be found.) ❑ Browsing source on the ***Work with Object History*** panel, using option **1** (***Browse current source***) and option **2** (***Browse archived source***). ❑ Browsing source on a worklist, using option **35** (***Browse source***). ❑ Browsing source anywhere else that TURNOVER® for iSeries v100 includes an option for doing so.
***CHGOBJ** **(Change Object Attributes)**	❑ Setting object attributes during a form promotion.
***CHGOBJOWN** **(Change Object Owner)**	❑ Setting object ownership (in primary and explode libraries) during a form promotion. ❑ Setting object ownership in temporary distribution libraries.
***CHKOBJ** **(Check for Object)**	❑ Setting the **Obj Y**/**N** flag on a worklist. ❑ Error checking when the promotion method is **MO** (Move Object), **CD** (Create Duplicate Object), **CSMO** (Copy Source, Move Object), or **CSCD** (Copy Source, Create Duplicate Object). ❑ Checking for an object anywhere that TURNOVER® for iSeries v100 needs to determine if an object exists. (There are many such places.)
***CHKSRC** **(Check for Object's Source)**	❑ Setting the **Src Y**/**N** flag on a worklist. ❑ Error checking when the promotion method is **CS** (Copy Source), **CSMO** (Copy Source, Move Object), or **CSCD** (Copy Source, Create Duplicate Object). ❑ Checking for an object's source anywhere that TURNOVER® for iSeries v100 needs to determine if source exists. (There are

	many such places.)

Method	In TURNOVER® for iSeries v100, this method is used for …
***CPYSRCF** **(Copy Object's Source)**	❑ Copying source during checkout. ❑ Copying source during a form promotion. ❑ Copying source when changing a checkout record. ❑ Copying source to the distribution library. ❑ Copying source during receive of distribution.
***CRTDUPOBJ** **(Create Duplicate Object)**	❑ Duplicating an object that has a non-source object type during checkout. ❑ Duplicating an object during a form promotion. ❑ Duplicating an object to the distribution library. ❑ Duplicating an object to the explode library.
***DLTOBJ** **(Delete Object)**	❑ Deleting an object during a form recovery. ❑ Deleting a from-object during a form promotion. ❑ Deleting an existing object before a worklist compile.
***DLTSRC** **(Delete Object's Source)**	❑ Deleting source when changing a checkout record. ❑ Deleting source during a form recovery. ❑ Deleting a from-source during a form promotion. ❑ Deleting source for a delete line ("D") on a form.
***EDTSRC** **(Edit Object's Source)**	❑ Editing source on a worklist, using option **32** (***Edit source***).
***GRTOBJAUT** **(Grant Object Authorities)**	❑ Setting object authorities (in primary and explode libraries) during a form promotion.
***MISSING** **(Create Missing Objects for Object Type)**	❑ Creating missing items on the ***Work with Application Definitions*** panel, using option **14** (***Create missing items***).
***MOVOBJ** **(Move Object)**	❑ Moving out an original object during a form promotion. ❑ Moving back an original object during a form recovery. ❑ Moving an object from the TnnnnnnnS library to the Thhmmssjjj library during distribution. ❑ Moving an object during a form promotion, using method **MO** (Move Object) or **CSMO** (Copy Source, Move Object). ❑ Moving an object from the Thhmmssjjj library to the

	TnnnnnnnS library during receive of distribution.
***RTVOBJD (Retrieve Object Description)**	❑ Returning data in a number of instances. For more information, read the description of this method on page 13.

STEP 3: CREATE CM METHOD PROGRAMS

This section describes the method program parameters, which are the same for all CM method programs. It also describes how you can use several TURNOVER® for iSeries v100 files and commands to simplify parsing parameters passed to and from the CM method programs.

All CM method programs must accept these four parameters:

Parameter	Length	Data Type	Description
Method Name	10	Character	The name of the CM method. This is the first parameter for all method programs. (See the table beginning on page 9 for a list and descriptions of methods.)
Object Data	2000	Character	Information about the object for which you are executing the method. For a detailed description of this parameter, see ***Object data parameter*** below.
Return Error Flag	1	Character	Specifies whether or not TURNOVER® for iSeries v100 needs to perform error handling for the method program and send informational messages to the calling program. The method programs must return a **Y** if it cannot successfully execute the method. For more information, see ***Error handling and messaging*** on page 21.
Return Data	2000	Character	Information returned by the method program. While only three CM methods (***ARCHIVE**, ***MISSING**, and ***RTVOBJD**) require data to be passed back to the calling program, all method programs must accept this parameter. For more information, see ***Return data parameter*** on page 21.

Object data parameter

The ***Object data*** parameter is the second parameter for all CM method programs. This parameter has 2000 positions, and not all of its data elements are relevant to all of the CM methods. The format of this parameter and the methods for which each data element is relevant follows:[3]

Data Element	Length	Data Type	Relevant CM Methods
Form number	7	Packed decimal	None
Line number	3	Packed decimal	None
Object type code	10	Character	All
Object type	10	Character	All
Promotion method	5	Character	None
Target object	10	Character	All
Target object library	10	Character	All
Target source file	10	Character	*CHKSRC, *CPYSRCF, *DLTSRC, *EDTSRC, *BROWSE, *CMPSRC, *ARCHIVE
Target source library	10	Character	*CHKSRC, *CPYSRCF, *DLTSRC, *EDTSRC, *BROWSE, *CMPSRC, *ARCHIVE
Target source member	10	Character	*CHKSRC, *CPYSRCF, *DLTSRC, *EDTSRC, *BROWSE, *CMPSRC, *ARCHIVE
From object	10	Character	*MOVOBJ, *CRTDUPOBJ
From object library	10	Character	*MOVOBJ, *CRTDUPOBJ
From source file	10	Character	*CPYSRCF, *ARCHIVE
From source file library	10	Character	*CPYSRCF, *ARCHIVE
From source member	10	Character	*CPYSRCF, *ARCHIVE
Programmer	10	Character	None
Reference	10	Character	None
Create parm	1	Character	*CHGOBJ
Create reference object	10	Character	*CHGOBJ
Create reference object library	10	Character	*CHGOBJ
Authority parm	1	Character	*CHGOBJOWN, *GRTOBJAUT
Authority reference object	10	Character	*CHGOBJOWN, *GRTOBJAUT
Authority reference object library	10	Character	*CHGOBJOWN, *GRTOBJAUT
Owner	10	Character	*CHGOBJOWN

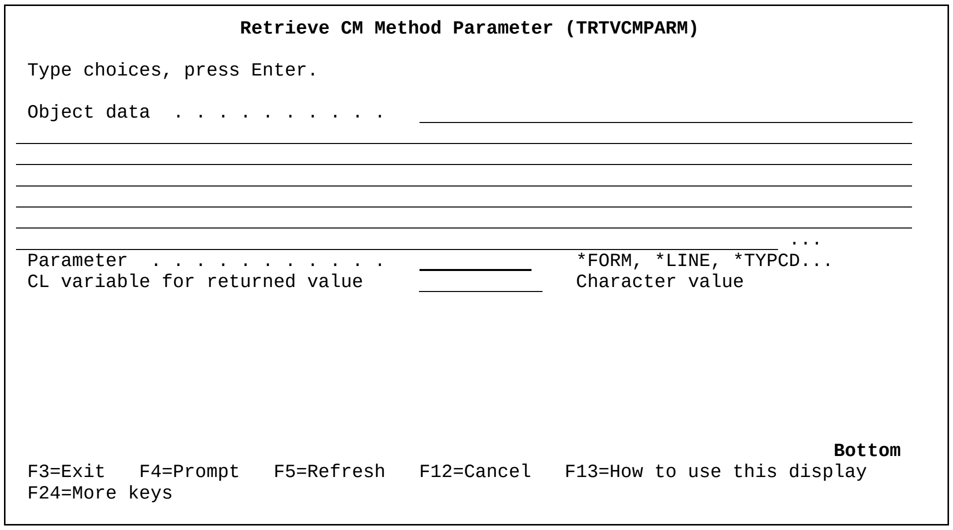
[3] A method program should not depend on, for a particular CM method, data elements that are not listed as being relevant to that method.

Parameter template file (TCMPARMS)

TURNOVER® for iSeries v100 provides a parameter template file (TCMPARMS) that has the field layout previously described in the table for the ***Object data*** parameter (see page 18). This file is provided for your convenience; you can have CM method programs use it to parse the incoming ***Object data*** parameter using an externally-defined data structure. See the sample CM schemes beginning on page 28.

TRTVCMPARM (Retrieve CM Method Parameter) command

You can use the **TRTVCMPARM** (**Retrieve CM Method Parameter**) command in a CM method program (CL program only) to parse the incoming ***Object data*** parameter into program variables.



```
                  Retrieve CM Method Parameter (TRTVCMPARM)

 Type choices, press Enter.

 Object data  . . . . . . . . . .   ____________________________________________
________________________________________________________________________________
________________________________________________________________________________
________________________________________________________________________________
________________________________________________________________________________
________________________________________________________________________________
____________________________________________________________________________ ...
 Parameter  . . . . . . . . . . .   __________    *FORM, *LINE, *TYPCD...
 CL variable for returned value      ___________   Character value







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

Descriptions of the **TRTVCMPARM** command parameters follow:

Object data

The ***Object data*** parameter (length is 2000; data type is Character) that TURNOVER® for iSeries v100 passed to the CM method.

Parameter

The data element (length is 10; data type is Character) you want retrieved from the ***Object data*** parameter. Valid values are:

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

CL variable for returned value

(Length is 10; data type is Character)

TRTVCMPARM command example

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

```
TRTVCMPARM OBJDTA(&OBJDTA) PARMNAME(*FOBJ) RTNVAR(&FROMOBJ)

TRTVCMPARM OBJDTA(&OBJDTA) PARMNAME(*FLIB) RTNVAR(&FROMLIB)

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

MYMOVCMD OBJ(&FROMLIB/&FROMOBJ) TOLIB(&TOLIB)
```

For additional examples of how you can use the **TRTVCMPARM** command in a method program, see the sample CM schemes beginning on page 28.

Error handling and messaging

All method programs must return a **Y** in the ***Return Error Flag*** parameter if the program cannot successfully execute its associated method. (For example, a ***CHKOBJ** method program should return **Y** if it cannot find the object.) This ensures that TURNOVER® for iSeries v100's processing continues appropriately.

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

Return data parameter

While all method programs must accept returned data (in the ***Return data*** parameter), only three CM methods require the method program to return data (other than the ***Return Error Flag***) to TURNOVER® for iSeries v100. The methods requiring returned data are ***RTVOBJD**, ***ARCHIVE**, and ***MISSING**. The required format for the returned data depends on which of these methods the method program is handling. Information about the returned data for these three CM methods is provided in the following sections.

*RTVOBJD (Retrieve Object Description) CM method

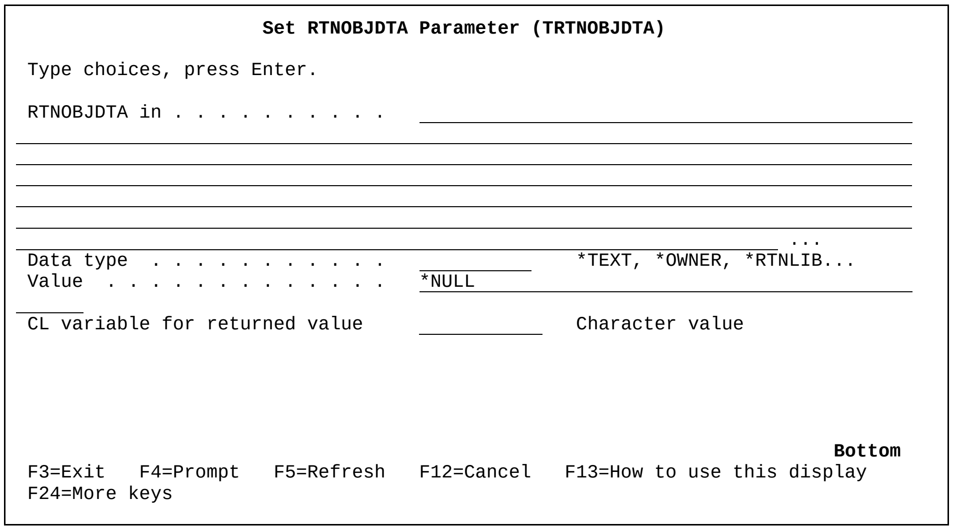
The data that you can have ***RTVOBJD** method programs return and the format of that data follows:

Data Element	Length	Data Type
Object text	50	Character
Owner	10	Character
Return library	10	Character
Object attribute	10	Character
Object user-defined attribute	10	Character
Create date	13	Character
Source file	10	Character
Source file library	10	Character
Source member	10	Character
Source date	13	Character
Object level	8	Character
Licensed program	16	Character
PTF	10	Character
APAR	10	Character

TURNOVER® for iSeries v100 provides a Return Object Description Data file (TRTNOBJDTA), which has the field layout described in the previous table. This file is provided for your convenience; you can have CM method programs use it to parse the returned data using an externally-defined data structure. (See the sample CM schemes beginning on page 28.)

TRTNOBJDTA (Set RTNOBJDTA Parameter) command

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



```
                         Set RTNOBJDTA Parameter (TRTNOBJDTA)

 Type choices, press Enter.

 RTNOBJDTA in . . . . . . . . . .    ____________________________________________
_______________________________________________________________________________
_______________________________________________________________________________
_______________________________________________________________________________
_______________________________________________________________________________
_______________________________________________________________________________
____________________________________________________________________ ...
 Data type  . . . . . . . . . . .    __________    *TEXT, *OWNER, *RTNLIB...
 Value  . . . . . . . . . . . . .    *NULL_______________________________________
______
 CL variable for returned value      ___________   Character value




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

TRTNOBJDTA command example

A ***RTVOBJD** method program for setting text and source data elements of the returned data might include this section of code:[4]

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

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

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

[4] It is not necessary to set *all* elements for *all* object types.

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

For additional examples of how you can use the **TRTNOBJDTA** command in a method program, see the sample CM schemes on page 28.

*MISSING (Create Missing Objects) CM method

TURNOVER® for iSeries v100 uses the ***MISSING** CM method when it checks an application or application level for missing items. When TURNOVER® for iSeries v100 checks for missing objects, it examines all the type code definitions for the application or level.

TURNOVER® for iSeries v100's default handling of the ***MISSING** CM method is to check, in the type code definition, for these objects for each type code:

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

For each of the above objects it does not find, TURNOVER® for iSeries v100 returns a creation command (**CRTLIB** . . ., **CRTSRCPF** . . ., **CRTREFOBJ** . . .) for the missing object. TURNOVER® for iSeries v100 displays the creation command in a panel selection list and executes the command if the user selects it.

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

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

The data elements from the ***Object data*** parameter (see ***Object data parameter*** on page 18) that are significant for the ***MISSING** CM method are:

Data Element	Description
Target object	The name of the library, source file, or reference object the method program will check.
Target object library	The library of the target object (library, source file, or reference object).
Object type	LIBRARY, SRC-PF, or REFOBJ.

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

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

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

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

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

The CM method program must check for the existence of the object, using whatever means is appropriate for the object type. If the program does not find the object, it must use the ***Return data*** parameter (see ***Return data parameter*** on page 21) to return the command string TURNOVER® for iSeries v100 will use to create the object. If the object does not need to be created, then you should have the program return blanks in the ***Return data*** parameter.

*ARCHIVE (Archive Object’s Source) CM method

TURNOVER® for iSeries v100 uses an ***ARCHIVE** method program to copy the archived source from the “source file” in the form archive library (T*nnnnnnn*D) to the archive “source file.”

Note: These files are probably not actual source files. If they were, it would not be necessary to override this method. (See **Important Information** on page 25.)

The data elements from the ***Object data*** parameter (see ***Object data parameter*** on page 18) that are significant for the ***ARCHIVE** CM method are:

Data Element	**Description**
From source file	The “source file” from which the source must be copied. (This will be the target source file on the form line.)
From source library	The T*nnnnnnn*D library.
From source member	The source “member” from which the source must be copied. (This will be the target source member on the form line.)
Target source file	The name of the source file specified as the ***Source archive file*** in the application definition.
Target source library	The name of the library specified as the ***Source archive file library*** in the application definition.
Target source member	The M*nnnnnnnnn* alias generated by TURNOVER® for iSeries v100.

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

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

TRTVCMPARM OBJDTA(&OBJDTA) PARMNAME(*FSRCL) RTNVAR(&FSRCL)

TRTVCMPARM OBJDTA(&OBJDTA) PARMNAME(*FSRCM) RTNVAR(&FSRCM)

TRTVCMPARM OBJDTA(&OBJDTA) PARMNAME(*TSRCF) RTNVAR(&TSRCF)

TRTVCMPARM OBJDTA(&OBJDTA) PARMNAME(*TSRCL) RTNVAR(&TSRCL)

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

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

Important Information

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

If the appropriate place is NOT the same as the Target source file, Target source file library, and Target source member data elements that were passed to the method program in the ***Object data*** parameter (as described on page 24), then the method program must use the ***Return data*** parameter (see ***Return data parameter*** on page 21) to return the actual values that identify where the “source” was archived. The format of the returned data must be:

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

TURNOVER® for iSeries v100 uses the returned data to correctly update TURNOVER® for iSeries v100’s archive history database. TURNOVER® for iSeries v100 then knows where the archived source resides if a request is made to browse or copy it.

The sample CM method programs that are shipped with TURNOVER® for iSeries v100 include an ***ARCHIVE** method program that demonstrates this concept.

CM method program considerations

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

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

There are several situations when TURNOVER® for iSeries v100 creates libraries and copies or moves source and objects to those libraries. For example, TURNOVER® for iSeries v100 creates T*nnnnnnn*D archive libraries when forms are run. TURNOVER® for iSeries v100 moves existing versions of objects that are on the form to that library before creating the new version. Also, if there is source for the object type, TURNOVER® for iSeries v100 copies the existing version of that source to the T*nnnnnnn*D archive library. When a form is distributed, TURNOVER® for iSeries v100 creates a work library and duplicates objects into that library, which it then saves. When a distributed form is received, TURNOVER® for iSeries v100 again moves and copies objects and source into the T*nnnnnnn*S from libraries.

Consequently, when a scheme uses a method program to handle the ***MOVOBJ**, ***CRTDUPOBJ**, or ***CPYSRCF** methods, you must keep an important consideration in mind when writing the program. That is, the method program must be capable of moving or duplicating objects and copying source into an empty library. The program must do whatever is necessary to establish the target environment for the move or copy. For example, if records in one or more database files represent an object type, then the method program must automatically create those database files in the target library if they do not exist there already. For examples, review the source code for the sample method programs included with the sample CM schemes beginning on page 28.

Relationship between the TUSRTYPF file and CM schemes

The TUSRTYPEF (TURNOVER® for iSeries v100 User-defined Type) file, serves these purposes:

- If TURNOVER® for iSeries v100 finds a record in file TUSRTYPF for the object type (for example, ***JDEDW**), it bypasses operations (such as changing owner, setting authorities, and so forth) that are not appropriate for the object type, when a form runs. By doing so, TURNOVER® for iSeries v100 does not generate errors (for example, ***"*JDEDW not valid for parameter OBJTYPE"***) that would result in form warnings.

- Prior to Release 5.3, when you added or changed a type code definition, you could supply an object type only if it was a valid iSeries object type (that is a valid value for the **CHKOBJ** parameter **OBJTYPE**) or if there was a record in file TUSRTYPF for the object type.

UNICOM Systems, Inc. Note

Object types that you will manage using CM schemes should NOT have records in file TUSRTYPF. To minimize impact on existing interfaces and rare instances of TURNOVER® for iSeries v100 hard-coding to accommodate special object types, TUSRTYPF continues to serve the same purpose as with previous releases (see the bulleted items on the previous page 26) during the running of a form. If there is a record in file TUSRTYPF, TURNOVER® for iSeries v100 will still bypass operations such as changing authorities, setting owners, and so forth, even if the object type uses a CM scheme.

STEP 4: CREATE TYPE CODES TO USE A CM SCHEME

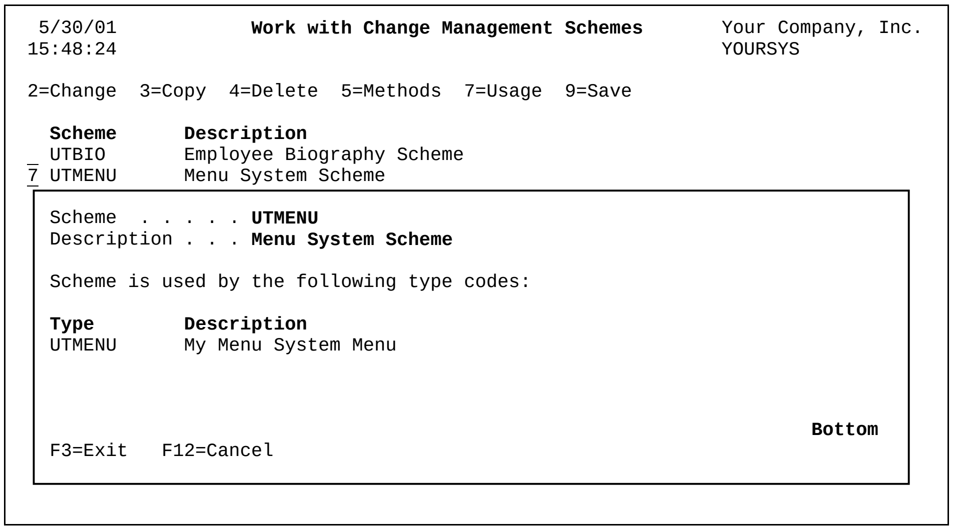
Once you create a CM scheme and override CM methods (as necessary), you can create new TURNOVER® for iSeries v100 type codes or change existing TURNOVER® for iSeries v100 type codes to use the new scheme. For information about how to do this, see ***Chapter 8: Utility Menu*** (specifically the ***Working with TURNOVER® for iSeries v100 Type Code Definitions*** section) in the ***TURNOVER® for iSeries v100 User Guide***.

OTHER SCHEME OPERATIONS

You can perform several useful operations with existing schemes. These operations are described in this section.

Displaying a CM scheme's usage

You might want to see which type codes will be affected if you change a scheme or method program. To display a list of TURNOVER® for iSeries v100 global type codes that are currently using a CM scheme, select a scheme on the ***Work with Change Management Schemes*** panel with option **7**. A pop-up panel appears:



```
 5/30/01           Work with Change Management Schemes       Your Company, Inc.
15:48:24                                                     YOURSYS

2=Change  3=Copy  4=Delete  5=Methods  7=Usage  9=Save

  Scheme      Description
_ UTBIO       Employee Biography Scheme
7 UTMENU      Menu System Scheme
 ┌──────────────────────────────────────────────────────────────────────────────┐
 │Scheme  . . . . . UTMENU                                                      │
 │Description . . . Menu System Scheme                                          │
 │                                                                              │
 │Scheme is used by the following type codes:                                   │
 │                                                                              │
 │Type        Description                                                       │
 │UTMENU      My Menu System Menu                                               │
 │                                                                              │
 │                                                                              │
 │                                                                     Bottom   │
 │F3=Exit   F12=Cancel                                                          │
 └──────────────────────────────────────────────────────────────────────────────┘
```

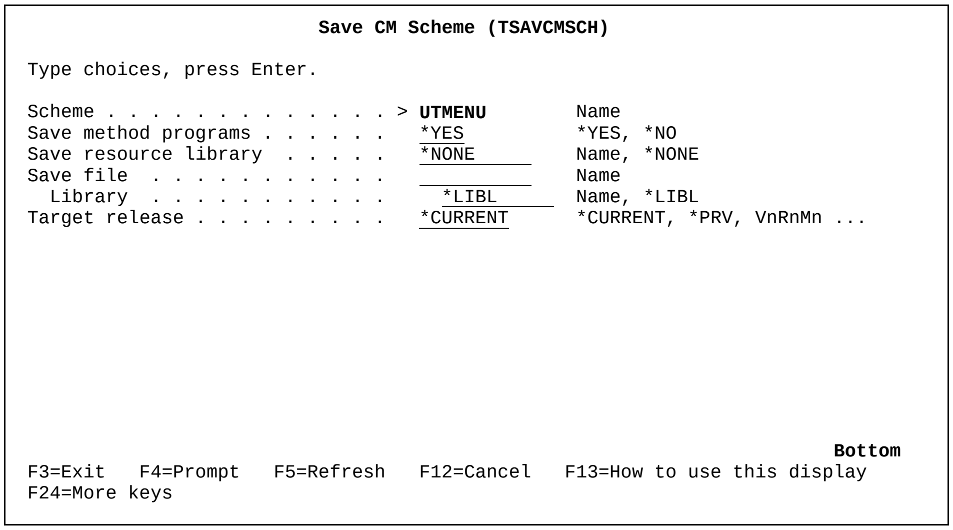
This pop-up panel displays a list of type codes that use the scheme you selected.

Saving a CM scheme

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

Note: This command (coupled with Restore a CM scheme; see page 30) is useful for exporting and importing your CM Schemes to other iSeries systems that need them. This is necessary if the application definition in which you have created this scheme and its associated Type Code also exists on remote (i.e. live production) machines to which you are distributing this Type Code's objects.

To save a CM scheme, select a scheme on the ***Work with Change Management Schemes*** panel with option **9** (or type **TSAVCMSCH** on a command line and press **Enter**). A ***Save CM Scheme*** panel appears:



```
                          Save CM Scheme (TSAVCMSCH)

Type choices, press Enter.

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










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

This panel displays the name of the scheme you selected. Beneath that, it prompts you for the information TURNOVER® for iSeries v100 needs to save the scheme (and optionally its method programs and resource library) to a save file. Descriptions of the fields follow:

Save method programs

Specify whether or not you want TURNOVER® for iSeries v100 to also save the CM method programs the scheme uses. Your choices are:

***YES**	Specify this to have TURNOVER® for iSeries v100 save the CM method programs the scheme uses.
***NO**	Specify this if you do not want TURNOVER® for iSeries v100 to save the CM method programs the scheme uses.

Save resource library

Specify whether or not you want TURNOVER® for iSeries v100 to also save the scheme's resource library. (The resource library contains any additional objects, such as files, called programs, commands, and so forth, required by the scheme's method programs.) Your choices are:

Name	Type the name of the resource library to have TURNOVER® for iSeries v100 save it. (TURNOVER® for iSeries v100 saves the entire library.)
***NONE**	Specify this if you do not want TURNOVER® for iSeries v100 to save the resource library.

Save file
Save file Library

Supply the name of the save file and library to which you want TURNOVER® for iSeries v100 to save the scheme. Your choices are:

Name	Type the names of the save file and library where the file resides.
***LIBL**	Specify this for the library name to have TURNOVER® for iSeries v100 search for the save file you have specified using the current library list.

If the save file does not exist, TURNOVER® for iSeries v100 creates it. If the save file already exists, and it contains data, TURNOVER® for iSeries v100 displays a message such as ***"Save file UTMENUSAVF in <<library name>> already contains data. (C G)"***. On the Reply line, type **C** (to cancel the save) or **G** (to go ahead with the save).

Target release

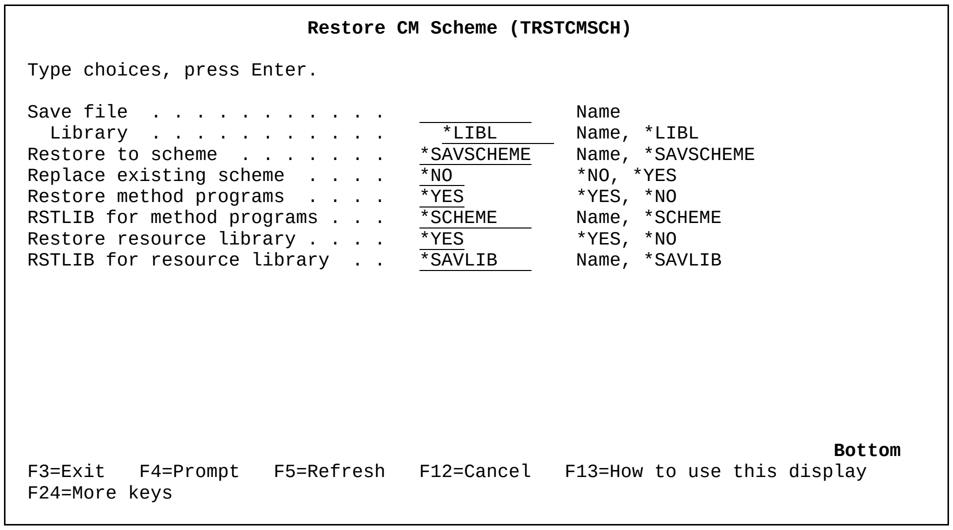
Specify the target system's OS/400 release level you want TURNOVER® for iSeries v100 to use for the save commands (**SAVLIB**, **SAVOBJ**, and so forth) when it saves the scheme, method programs, and resource library. Your choices are ***CURRENT** or **VnRnMn**.

Restoring a CM scheme

You can restore a previously saved CM scheme definition, and optionally restore its method programs and resource library (containing any additional objects required by the method programs), from a save file that was created with the **TSAVCMSCH** command.

Note: This command (coupled with Save a CM scheme; see page 30) are useful for exporting and importing your CM Schemes to other iSeries systems that need them. This is necessary if the application definition in which you have created this scheme and its associated Type Code also exists on remote (i.e. live production) machines to which you are distributing this Type Code's objects.

To restore a saved CM scheme, press **F8** on the ***Work with Change Management Schemes*** panel (or type **TRSTCMSCH** on a command line and press **Enter**). A ***Restore CM Scheme*** panel appears:



```
                         Restore CM Scheme (TRSTCMSCH)

Type choices, press Enter.

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







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

This panel prompts you for the information TURNOVER® for iSeries v100 needs to restore the scheme (and optionally its method programs and resource library) from a save file. Descriptions of the fields follow:

Save file
Save file library

Supply the name of the save file and library from which you want TURNOVER® for iSeries v100 to restore the scheme. (The save file you specify must have been created with the **TSAVCMSCH** command; see page 30.) Your choices are:

Name	Type the names of the save file and library where the save file resides.
***LIBL**	Specify this for the library name to have TURNOVER® for iSeries v100 to search for the save file you have specified using the current library list.

Restore to scheme

Supply the name of the scheme you want TURNOVER® for iSeries v100 to create or update. (If the scheme you specify here already exists, you must specify ***YES** for the next field, **Replace existing scheme**.) Your choices are:

Name	Type the name of the scheme.
***SAVSCHEME**	Specify this if the name of the scheme you want to restore is the same as the scheme that was saved.

Replace existing scheme

Specify whether or not you want TURNOVER® for iSeries v100 to overlay an existing scheme with the scheme it is restoring, if the two have the same name. Your choices are:

***NO**	Specify this to have TURNOVER® for iSeries v100 abort the command if the scheme it is restoring already exists (that is, the existing scheme has the same name as the scheme TURNOVER® for iSeries v100 is restoring).
***YES**	Specify this to have TURNOVER® for iSeries v100 overlay an existing scheme with the scheme it is restoring, if the two have the same name).

Restore method programs

Specify whether or not you want TURNOVER® for iSeries v100 to restore any CM method programs that were saved with the scheme. Your choices are:

***YES**	Specify this to have TURNOVER® for iSeries v100 restore the CM method programs.
***NO**	Specify this if you do not want TURNOVER® for iSeries v100 to restore the CM method programs.

RSTLIB for method programs

Supply the name of the library to which you want TURNOVER® for iSeries v100 to restore any CM method programs that were saved with the scheme. (TURNOVER® for iSeries v100 ignores this field if you specified ***NO** for the previous **Restore method programs** field.) Your choices are:

Name	Type a specific library name.
***SCHEME**	Specify this if you want TURNOVER® for iSeries v100 to restore CM method programs to the library specified for the scheme method.

Restore resource library

Specify whether or not you want TURNOVER® for iSeries v100 to restore any resource library that was saved with the scheme. The resource library contains any additional objects, such as files, called programs, commands, and so forth, required by the scheme's method programs. Your choices are:

***YES**	Specify this to have TURNOVER® for iSeries v100 also restore any resource library that was saved with the scheme.
***NO**	Specify this if you do not want TURNOVER® for iSeries v100 to restore any resource library that was saved with the scheme.

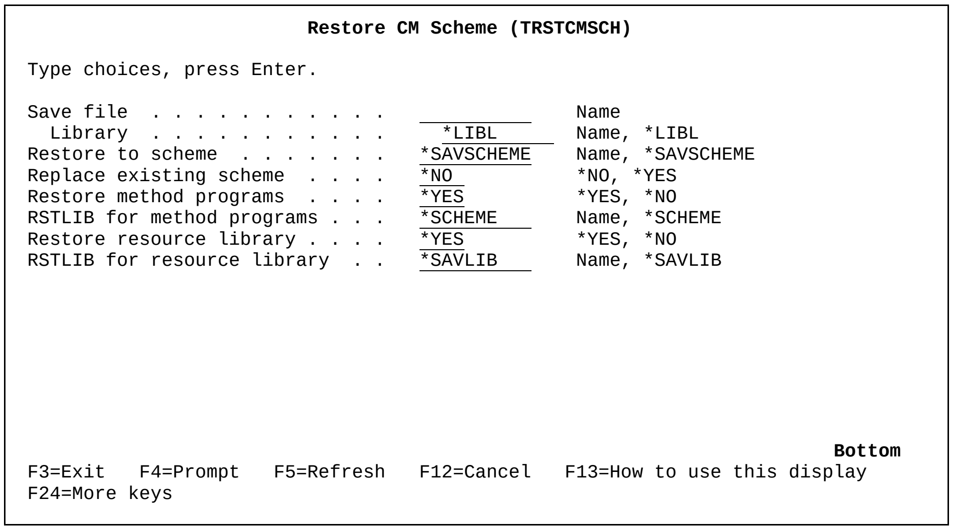
RSTLIB for resource library

Supply the name of the library to which you want TURNOVER® for iSeries v100 to restore the resource library that was saved with the scheme. (TURNOVER® for iSeries v100 ignores this field if you specified ***NO** for the previous **Restore resource library** field.) Your choices are:

Name	Type a specific library name.
***SAVLIB**	Specify this if you want TURNOVER® for iSeries v100 to restore the resource library from the same library to which it was saved.

Generating Source from a CM scheme

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



```
                         Restore CM Scheme (TRSTCMSCH)

Type choices, press Enter.

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









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

Descriptions of the fields follow:

Scheme

Supply the name of the scheme for which you would like this command to generate source.

Method

Supply the name of the method for which you would like this command to generate source. Your choices are:

***ALL** Specify this to have TURNOVER® for iSeries v100 abort the command if the scheme it is restoring already exists (that is, the existing scheme has the same name as the scheme TURNOVER® for iSeries v100 is restoring).

{Method Name} Select one or more of the 16 schemes for which this command is to generate source.

Source File, Library

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

Replace or Add

Whether you want to overlay existing source member or create new source member.Your choices are:

***REPLACE**	Overlay existing source code member’s source.
***ADD**	Create a new source member.

SAMPLE 1 – UTMENU SCHEME (MENU SYSTEM)

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

UTMENU – Menu Header File

Menu Name (Key)

Menu Description

Menu Owner (Retrieved using ***RTVOBJD** method; set using ***CHBOBJOWN** method)

Menu Active Y/N (Set using ***CHGOBJ** method)

Menu Source File (Retrieved using ***RTVOBJD** method; set by **CRTUTMENU** command)

Menu Source File Library (Retrieved using ***RTVOBJD** method; set by **CRTUTMENU** command)

Menu Source Member (Retrieved using ***RTVOBJD** method; set by **CRTUTMENU** command)

Menu Create Date (Retrieved using ***RTVOBJD** method; set by **CRTUTMENU** command)

Menu Source Date (not used)

UTMENUOPT – Menu Options File

Menu Name (Key)

Option Name (Key)

Option Description

Option Command to Execute

UTMENUAUT – Menu Authorized Users File

Menu Name (Key)

Authorized User (Key)

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

In addition to the UTMENU, UTMENUOPT, and UTMENUAUT files, where a menu's run-time definition is stored, another file, UTMENSRC, stores XML style "source" (the file is not actually a source physical file) members that TURNOVER® for iSeries v100 uses to construct menus using the **UTCRTMENU** command.

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

UNICOM Systems, Inc. Note

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

The restrictions imposed by the **UTCRTMENU** command include these:

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

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

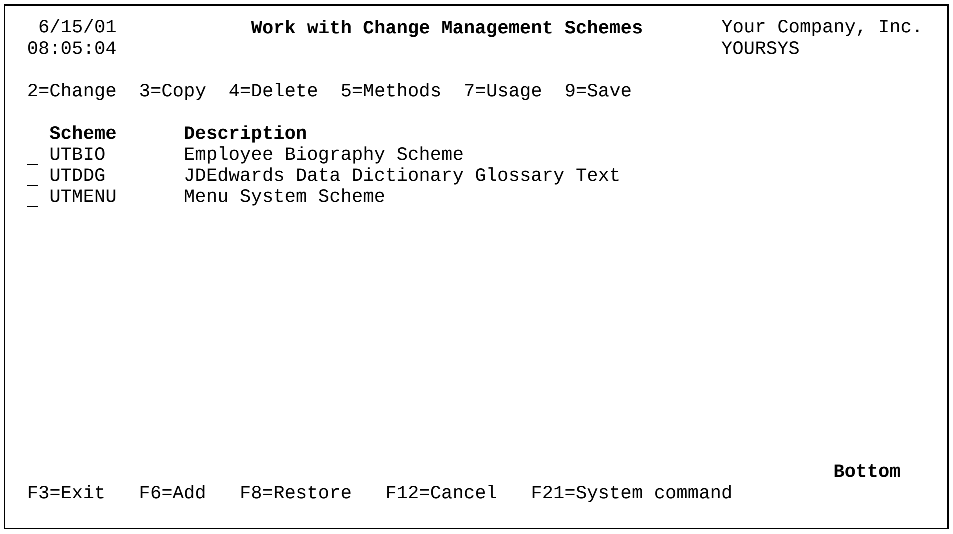
Follow the instructions (see next) for restoring the UTMENU sample CM scheme. Once you restore the scheme, review source code for the scheme's method programs, paying special attention to the tips and techniques described in the remainder of the UTMENU topic.

Restoring the UTMENU scheme

To restore the UTMENU scheme, you must perform these steps:

1. Sign on as a user profile with authority to restore libraries and TURNOVER® for iSeries v100 application definitions to the system.

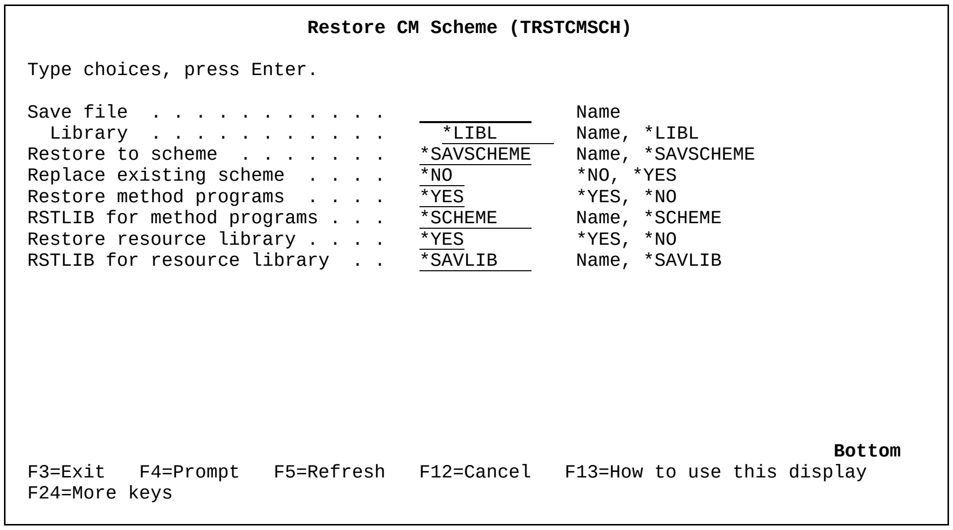
2. On a TURNOVER® for iSeries v100 command line, type **TWRKCMSCH** and press **Enter**. The ***Work with Change Management Schemes*** panel appears:



```
 6/15/01           Work with Change Management Schemes       Your Company, Inc.
08:05:04                                                     YOURSYS

2=Change  3=Copy  4=Delete  5=Methods  7=Usage  9=Save

  Scheme      Description
_ UTBIO       Employee Biography Scheme
_ UTDDG       JDEdwards Data Dictionary Glossary Text
_ UTMENU      Menu System Scheme











                                                                       Bottom
F3=Exit   F6=Add   F8=Restore   F12=Cancel   F21=System command
```

3. If the UTMENU scheme is already listed on this panel, proceed to step 6.

4. Press **F8** (***Restore***). The ***Restore CM Scheme (TRSTCMSCH)*** panel appears:



```
                         Restore CM Scheme (TRSTCMSCH)

Type choices, press Enter.

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









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

On this panel, type **UTMENUSAVF** for the save file name. Accept the default values for the other parameters (the save file is in the TURNOVER® for iSeries v100 product library). Press **Enter**.

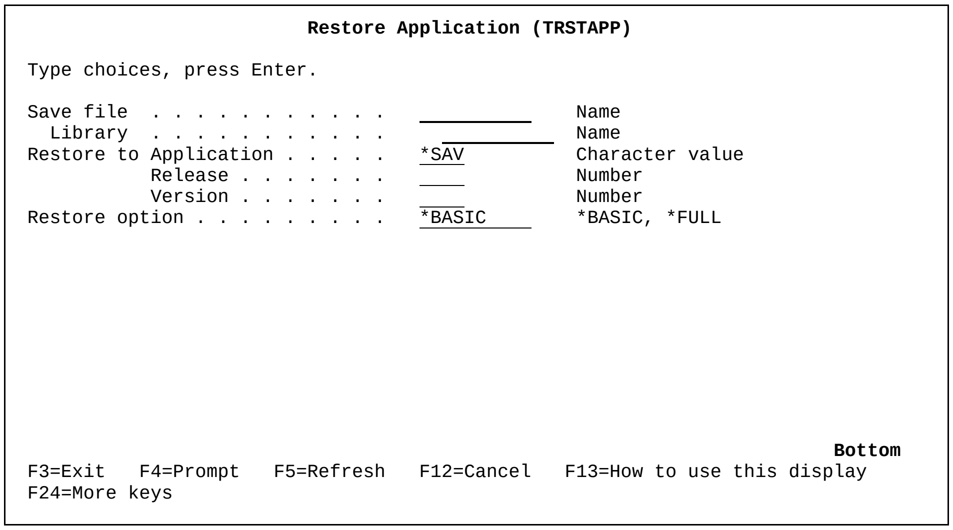
TURNOVER® for iSeries v100 then restores these items:

a) The UTMENU scheme definition.

b) Library UTMENUPRD (the resource library), which contains the menu definition and menu source files (described on pages 28 and 38), the **UTCRTMENU** command (and command processing programs), and the UTMENU CM method programs.

5. Select TURNOVER® for iSeries v100 Main Menu option **1** (***Work with Application Definitions***). If application UT is already on the ***Work with Application Definitions panel*** (you might have to page down to see it), proceed to step 8.

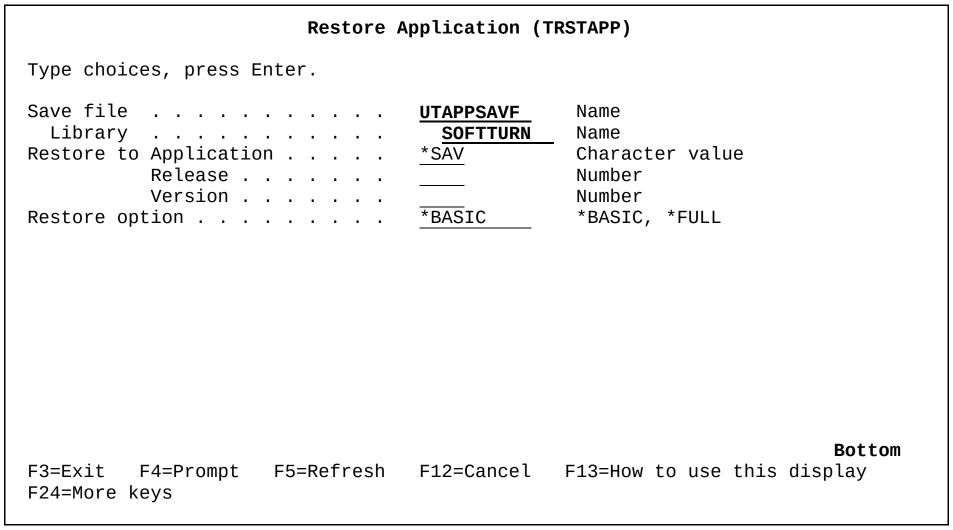
6. On a TURNOVER® for iSeries v100 command line, type **TRSTAPP**. The ***Restore Application (TRSTAPP)*** panel appears:



```
                        Restore Application (TRSTAPP)

 Type choices, press Enter.

 Save file  . . . . . . . . . . .   __________    Name
   Library  . . . . . . . . . . .     __________  Name
 Restore to Application . . . . .   *SAV          Character value
            Release . . . . . . .   ____          Number
            Version . . . . . . .   ____          Number
 Restore option . . . . . . . . .   *BASIC        *BASIC, *FULL









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

On this panel, type **UTAPPSAVF** for the save file name. For the save file library, type the name of your TURNOVER® for iSeries v100 product library (normally SOFTTURN).

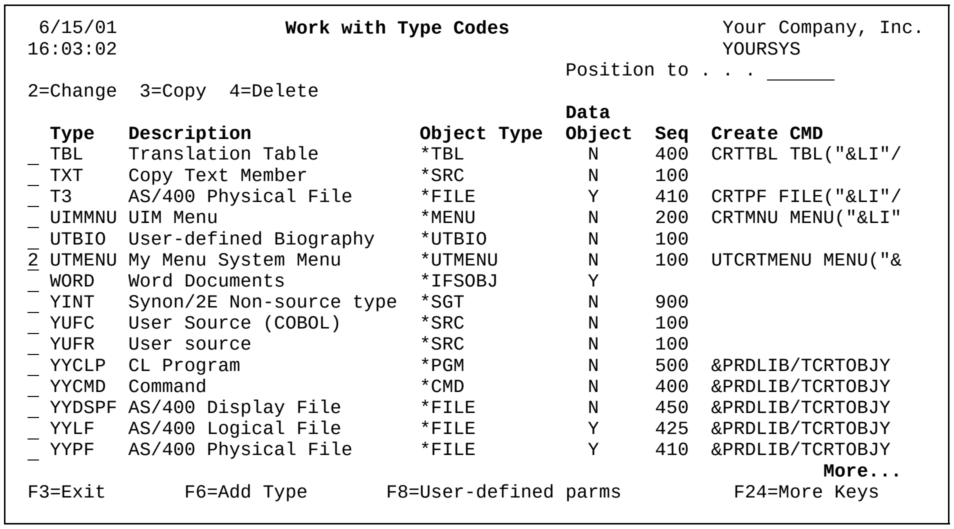
The panel now looks like this:



```
                         Restore Application (TRSTAPP)

Type choices, press Enter.

Save file  . . . . . . . . . . .   UTAPPSAVF      Name
  Library  . . . . . . . . . . .     SOFTTURN     Name
Restore to Application . . . . .   *SAV           Character value
           Release . . . . . . .   ____           Number
           Version . . . . . . .   ____           Number
Restore option . . . . . . . . .   *BASIC         *BASIC, *FULL










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

Accept the default values for the other parameters and press **Enter**. TURNOVER® for iSeries v100 restores the UT application definition as well as the UTMENU global type code definition.

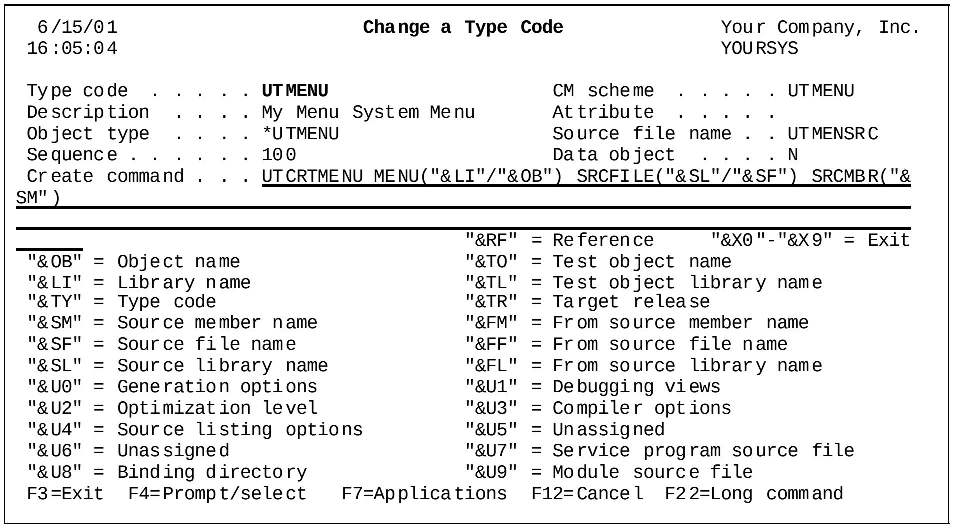
7. Select TURNOVER® for iSeries v100 Main Menu option **8** (***Utility Menu***) and then option **4** (***Work with type code definitions***). The ***Work with Type Codes*** panel appears. Type **2** next to the UTMENU type code:



```
 6/15/01              Work with Type Codes                 Your Company, Inc.
16:03:02                                                   YOURSYS
                                                 Position to . . . ______
2=Change  3=Copy  4=Delete
                                                 Data
  Type   Description             Object Type   Object  Seq  Create CMD
_ TBL    Translation Table       *TBL            N     400  CRTTBL TBL("&LI"/
_ TXT    Copy Text Member        *SRC            N     100
_ T3     AS/400 Physical File    *FILE           Y     410  CRTPF FILE("&LI"/
_ UIMMNU UIM Menu                *MENU           N     200  CRTMNU MENU("&LI"
_ UTBIO  User-defined Biography  *UTBIO          N     100
2 UTMENU My Menu System Menu     *UTMENU         N     100  UTCRTMENU MENU("&
_ WORD   Word Documents          *IFSOBJ         Y
_ YINT   Synon/2E Non-source type *SGT           N     900
_ YUFC   User Source (COBOL)     *SRC            N     100
_ YUFR   User source             *SRC            N     100
_ YYCLP  CL Program              *PGM            N     500  &PRDLIB/TCRTOBJY
_ YYCMD  Command                 *CMD            N     400  &PRDLIB/TCRTOBJY
_ YYDSPF AS/400 Display File     *FILE           N     450  &PRDLIB/TCRTOBJY
_ YYLF   AS/400 Logical File     *FILE           Y     425  &PRDLIB/TCRTOBJY
_ YYPF   AS/400 Physical File    *FILE           Y     410  &PRDLIB/TCRTOBJY
                                                                     More...
F3=Exit       F6=Add Type       F8=User-defined parms          F24=More Keys
```

Press **Enter**.

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



```
 6/15/01                       Change a Type Code             Your Company, Inc.
16:05:04                                                      YOURSYS

Type code  . . . . . UTMENU                      CM scheme  . . . . . UTMENU
Description  . . . . My Menu System Menu         Attribute  . . . . .
Object type  . . . . *UTMENU                     Source file name . . UTMENSRC
Sequence . . . . . . 100                         Data object  . . . . N
Create command . . . UTCRTMENU MENU("&LI"/"&OB") SRCFILE("&SL"/"&SF") SRCMBR("&
SM")

                                        "&RF" = Reference     "&X0"-"&X9" = Exit
"&OB" = Object name                     "&TO" = Test object name
"&LI" = Library name                    "&TL" = Test object library name
"&TY" = Type code                       "&TR" = Target release
"&SM" = Source member name              "&FM" = From source member name
"&SF" = Source file name                "&FF" = From source file name
"&SL" = Source library name             "&FL" = From source library name
"&U0" = Generation options              "&U1" = Debugging views
"&U2" = Optimization level              "&U3" = Compiler options
"&U4" = Source listing options          "&U5" = Unassigned
"&U6" = Unassigned                      "&U7" = Service program source file
"&U8" = Binding directory               "&U9" = Module source file
F3=Exit  F4=Prompt/select   F7=Applications  F12=Cancel  F22=Long command
```

8. Return to TURNOVER® for iSeries v100 Main Menu option **1** (***Work with Application Definitions***) and type option **14** (***Create missing items***) next to the UT application. Unless they have been created previously, you will be prompted to create library UTMENUDEV (development library) and file UTMENUDEV/UTMENSRC (development source). Because UTMENSRC is not actually a source file, we have used the UTMENU CM scheme (***MISSING** method) to override (from **CRTPF*xxxx*** to **CRTDUPOBJ*xxxxx***) the create command for the source file. Press **Enter** to create the missing objects.

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

UTMENU scheme objects

The source code for all the objects that were restored into library UTMENUPRD when you restored the UTMENU CM scheme can be found in source files ADDSSRC (PFs), ACMDSRC (CMD), ACLSRC (CLLEs) and ARPGLESRC (RPGLEs) in the TURNOVER® for iSeries v100 product library. The restored objects include:

- UTMENU, UTMENOPT, UTMENAUT, UTMENSRC. (These are the menu definition files described on page 28.)
- UTMENARC – The source archive file for the UTMENU scheme. The format is identical to UTMENSRC. (See the description of method program UTARCHIVE later in this section for a description of how this file is used.)
- UTCRTMENU (CMD, CLLE) and UTCRTMENUR (RPGLE) – The command and command processing programs used to construct UTMENU menus.
- UTARCHIVE (CLLE) – ***ARCHIVE** method program.
- UTCHGOBJ (CLLE) and UTCHGOBJR (RPGLE) – ***CHGOBJ** method program.
- UTCHGOWN (CLLE) and UTCHGOWNR (RPGLE) – ***CHGOBJOWN** method program.
- UTCHKOBJ (CLLE) and UTCHKOBJR (RPGLE) – ***CHKOBJ** method program.
- UTCHKSRC (CLLE) – ***CHKSRC** method program.
- UTCMPSRC (CLLE) – ***CMPSRC** method program.
- UTCPYSRCF (CLLE) – ***CPYSRCF** method program.
- UTDLTOBJ (CLLE) and UTDLTOBJR (RPGLE) – ***DLTOBJ** method program.
- UTDLTSRC (CLLE) – ***DLTSRC** method program.
- UTEDTBRW (CLLE) – ***EDTSRC** and ***BROWSE** method program.
- UTGRTAUT (CLLE) and UTGRTAUTR (RPGLE) – ***GRTOBJAUT** method program.
- UTMISSING (CLLE) – ***MISSING** method program.
- UTMOVDUP (CLLE) and UTMOVDUPR (RPGLE) – ***MOVOBJ** and ***DRTDUPOBJ** method programs.
- UTRTVOBJD (CLLE) and UTRTVOBJDR (RPGLE) – ***RTVOBJD** method program.

CM method program parameters

All CM method programs must accept the same four parameters (see ***Step 3: Create CM Method Programs*** beginning on page 17). You can use the parameter definitions for the UTMENU method programs as templates for creating your own method programs.

Using the TRTVCMPARM command in a CL method program

The second parameter passed to all CM method programs is a 2000-character string from which all information needed to execute the method can be retrieved (object name, from library, target library, and so forth). You can use the TURNOVER® for iSeries v100 command, **TRTVCMPARM**, to retrieve individual data elements from this 2000-character string. For examples of this, review the source code for any of the following UTMENU method programs:

UTARCHIVE, UTCHGOWN, UTCHKSRC, UTCMPSRC, UTCPYSRCF,

UTDLTOBJ, UTDLTSRC, UTEDTBRW, UTMISSING, UTMOVDUP, UTRTVOBJD

See the next topic for examples of how to retrieve data elements from an RPGLE program.

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

You can use the externally-defined TURNOVER® for iSeries v100 file, TCMPARMS, in an RPGLE program to create a data structure over the second parameter passed to all CM method programs (see previous paragraph) to easily parse the individual data elements from this 2000-character string. For examples of this technique, review the source code for any of the following UTMENU method programs (these programs are actually called by method programs):

UTCHGOBJR, UTCHKOBJR, UTGRTAUTR

Using the TRTNOBJDTA command in a CL method program

The ***RTVOBJD** (**Retrieve Object Description**) CM method program must return object description data using the fourth parameter, which is a 2000-character string that must be correctly formatted (that is, correct data elements in correct positions). You can use the TURNOVER® for iSeries v100 command, **TRTNOBJDTA**, to set the individual data elements to the appropriate values. For an example of this, review the source code for the UTMENU method program UTRTVOBJD.

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

You can use the externally-defined TURNOVER® for iSeries v100 file, TRTNOBJDTA, in an RPGLE program to create a data structure over the fourth parameter (see previous paragraph) to easily parse the individual data elements from this 2000-character string. For examples of this, review the source code for the UTMENU method program UTCHGOBJR (called by method program UTCHGOBJ).

Handling multiple CM methods in the same method program

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

Executing a CM method from another CM method

Sometimes you might find it useful to execute one CM method from within the method program that handles another CM method. You can use the TURNOVER® for iSeries v100 command, **TRUNCMMTH**, to do this. For an example, review the source code for the UTMENU program UTMISSING (***MISSING** method), which uses the **TRUNCMMTH** command to execute the ***CHKOBJ** (**Check for Object**) method.

Overriding the source archive file from the *ARCHIVE method

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

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

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

Checking for missing objects with the *MISSING method

After you read the description of the ***MISSING** (**Create Missing Items**) method (see ****MISSING (Create Missing Objects) CM method*** on page 23), review the source code for the UTMENU method program UTMISSING as an example of how you use this method.

Retrieving the default scheme library from the TSCHEMELIB data area

When defining a CM scheme, one of the special values you can use for the method program library (for the scheme's default method program library or for an individual method's method program libraries, or for both) is ***SCHEMELIB**. This special value causes TURNOVER® for iSeries v100 to retrieve the library name from the TSCHEMELIB data area. (The TURNOVER® for iSeries v100 user or administrator is responsible for setting this data area to the appropriate library name). If a CM method program needs to call other programs, and you cannot assume that TURNOVER® for iSeries v100 will find those programs in the library list, you can use a convenient technique. Place those programs in the *SCHEMELIB library (the library found in TSCHEMELIB data area) and code calling method programs to retrieve the library of the called program from there. For an example of this technique, review the source code for these UTMENU method programs:

UTCHGOBJ, UTCHGOWN, UTCHKOBJ, UTGRTAUT, UTMOVDUP, UTRTVOBJD

File overrides and opens from CM method programs

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

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

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

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

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

SAMPLE 2 – UTBIO SCHEME (EMPLOYEE BIOGRAPHIES)

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

UTBIO – Employee Biography File

Department Code (Key)

Employee ID (Key)

Biography Text

Because the total length of the file's key is 20 (Department Code and Employee ID are each 10), you cannot identify an employee biography object using a 10-character object name such as TURNOVER® for iSeries v100's database supports. Consequently, this scheme uses aliases, which are stored in an alias file. (For additional details, see ***Using object aliases in CM schemes*** on page 53.) This file and its fields are:

UTBIOALIAS – Employee Biography Alias File

Alias (Key)

Department Code

Option Description

Employee ID

For retrieving an existing alias for an employee biography, there is a logical file over the alias file. This file and its fields are:

UTBIOALILF – Employee Biography Alias Logical File

Department Code (Key)

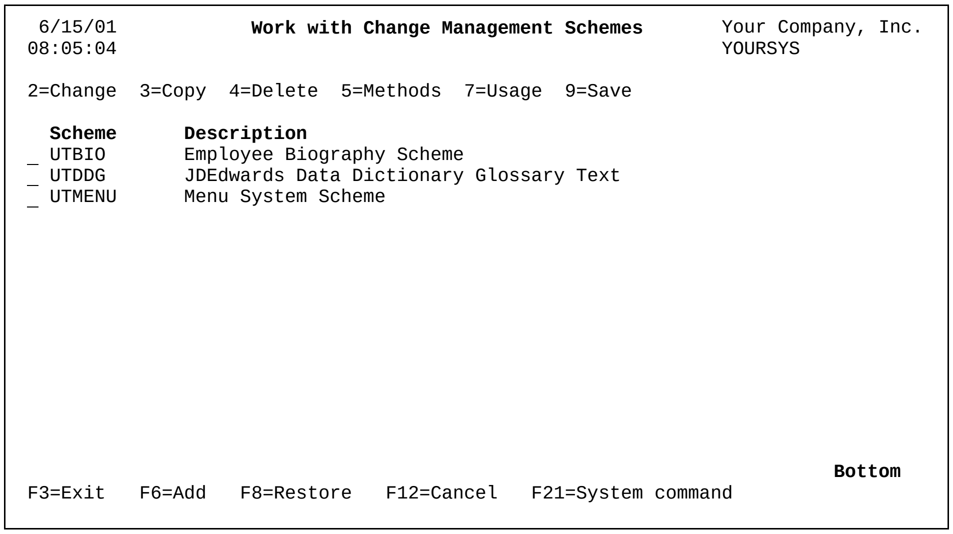
Employee ID (Key)

Restoring the UTBIO scheme

To restore the UTBIO scheme, you must perform these steps:

1. Sign on as a user profile with authority to restore libraries and TURNOVER® for iSeries v100 application definitions to the system.

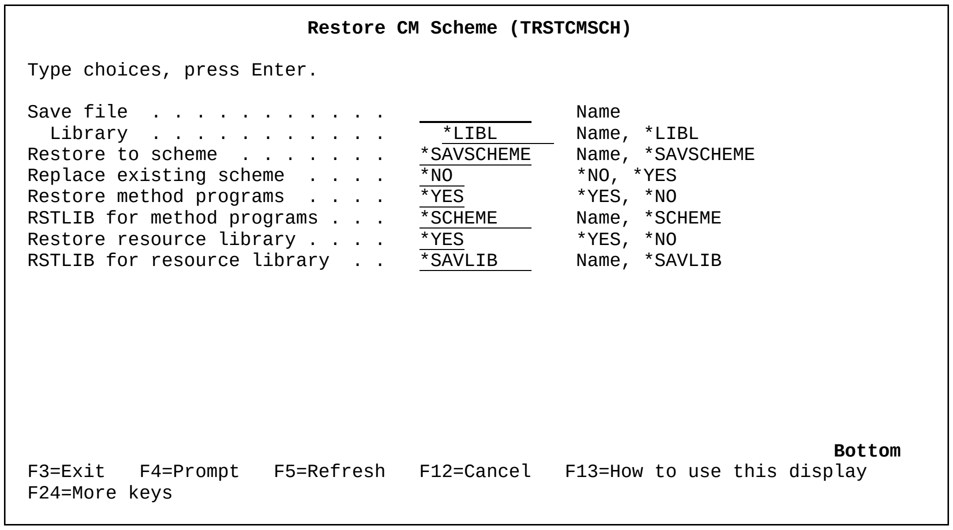
2. On a TURNOVER® for iSeries v100 command line, type **TWRKCMSCH** and press **Enter**. The ***Work with Change Management Schemes*** panel appears:



```
 6/15/01            Work with Change Management Schemes       Your Company, Inc.
08:05:04                                                      YOURSYS

2=Change  3=Copy  4=Delete  5=Methods  7=Usage  9=Save

  Scheme      Description
_ UTBIO       Employee Biography Scheme
_ UTDDG       JDEdwards Data Dictionary Glossary Text
_ UTMENU      Menu System Scheme










                                                                        Bottom
F3=Exit   F6=Add   F8=Restore   F12=Cancel   F21=System command
```

3. If the UTBIO scheme is already listed on this panel, proceed to step 6.

4. Press **F8** (**Restore**). The ***Restore CM Scheme (TRSTCMSCH)*** panel appears:



```
                        Restore CM Scheme (TRSTCMSCH)

Type choices, press Enter.

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







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

On this panel, type **UTBIOSAVF** for the save file name. Accept the default values for the other parameters (the save file is in the TURNOVER® for iSeries v100 product library). Press **Enter**.

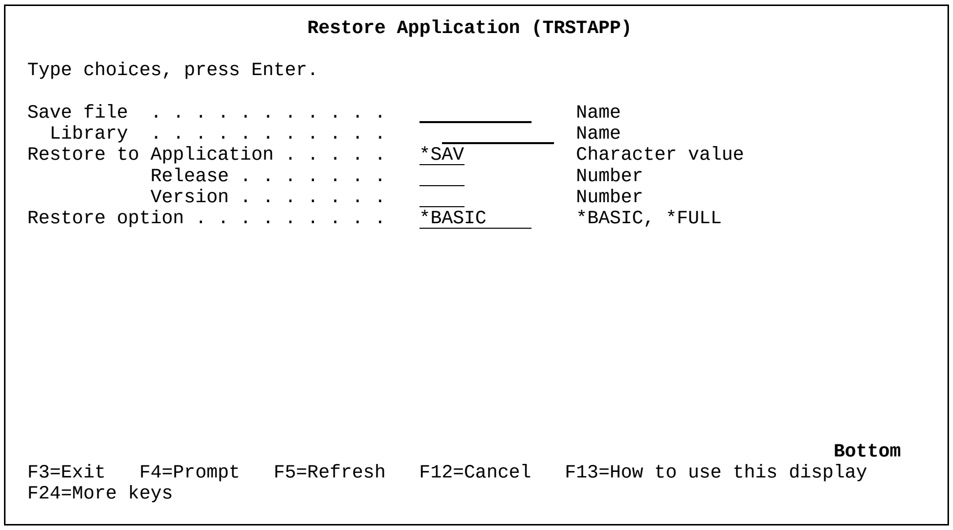
TURNOVER® for iSeries v100 then restores these items:

a) The UTBIO scheme definition.

b) Library UTBIOPRD (the resource library), which contains the employee biography and alias files (described on page 47) and the UTBIO CM method programs.

5. Select TURNOVER® for iSeries v100 Main Menu option **1** (***Work with Application Definitions***). If application UT is already on the ***Work with Application Definitions panel*** (you might have to page down to see it), proceed to step 8. (If you have restored the previous UTMENU sample, it will be there.)

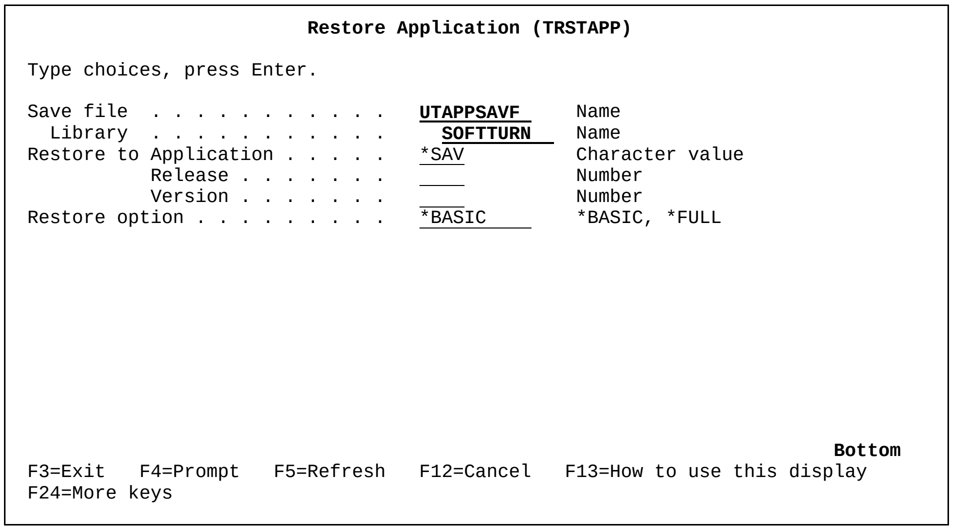
6. On a TURNOVER® for iSeries v100 command line, type **TRSTAPP**. The ***Restore Application (TRSTAPP)*** panel appears:



```
                          Restore Application (TRSTAPP)

Type choices, press Enter.

Save file  . . . . . . . . . . .   __________    Name
  Library  . . . . . . . . . . .     __________  Name
Restore to Application . . . . .   *SAV          Character value
           Release . . . . . . .   ____          Number
           Version . . . . . . .   ____          Number
Restore option . . . . . . . . .   *BASIC        *BASIC, *FULL










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

On this panel, type **UTAPPSAVF** for the save file name. For the save file library, type the name of your TURNOVER® for iSeries v100 product library (normally SOFTTURN).

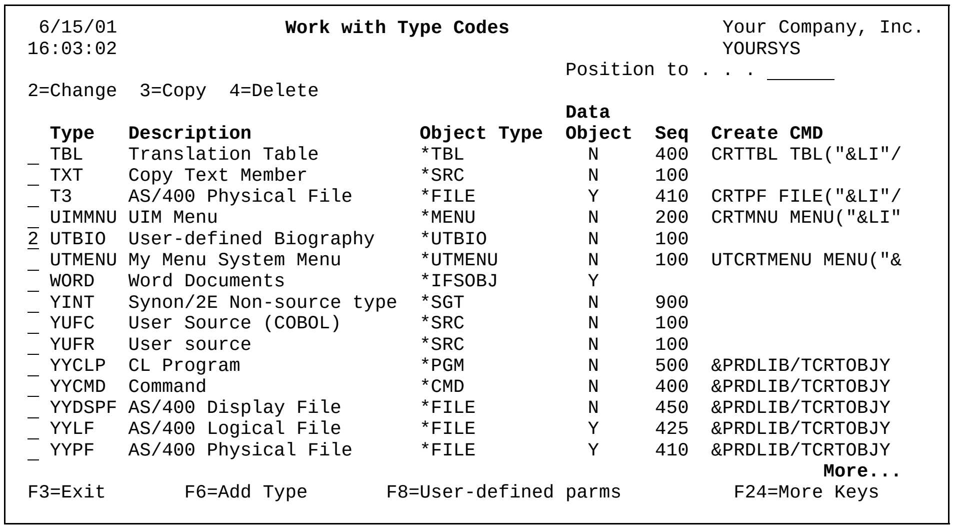
The panel now looks like this:



```
                          Restore Application (TRSTAPP)

Type choices, press Enter.

Save file  . . . . . . . . . . .   UTAPPSAVF     Name
  Library  . . . . . . . . . . .     SOFTTURN    Name
Restore to Application . . . . .   *SAV          Character value
           Release . . . . . . .   ____          Number
           Version . . . . . . .   ____          Number
Restore option . . . . . . . . .   *BASIC        *BASIC, *FULL










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

Accept the default values for the other parameters and press **Enter**. TURNOVER® for iSeries v100 restores the UT application definition as well as the UTBIO global type code definition.

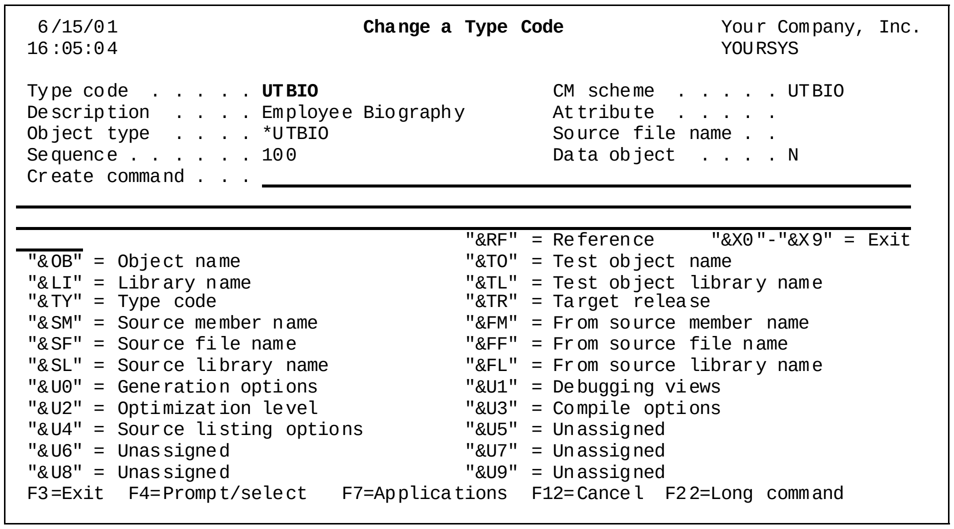
7. Select TURNOVER® for iSeries v100 Main Menu option **8** (***Utility Menu***) and then option **4** (***Work with type code definitions***). The ***Work with Type Codes*** panel appears. Type **2** next to the UTBIO type code:



```
 6/15/01              Work with Type Codes                 Your Company, Inc.
16:03:02                                                   YOURSYS
                                              Position to . . . ______
2=Change  3=Copy  4=Delete
                                              Data
  Type   Description               Object Type  Object  Seq  Create CMD
_ TBL    Translation Table         *TBL           N     400  CRTTBL TBL("&LI"/
_ TXT    Copy Text Member          *SRC           N     100
_ T3     AS/400 Physical File      *FILE          Y     410  CRTPF FILE("&LI"/
_ UIMMNU UIM Menu                  *MENU          N     200  CRTMNU MENU("&LI"
2 UTBIO  User-defined Biography    *UTBIO         N     100
_ UTMENU My Menu System Menu       *UTMENU        N     100  UTCRTMENU MENU("&
_ WORD   Word Documents            *IFSOBJ        Y
_ YINT   Synon/2E Non-source type  *SGT           N     900
_ YUFC   User Source (COBOL)       *SRC           N     100
_ YUFR   User source               *SRC           N     100
_ YYCLP  CL Program                *PGM           N     500  &PRDLIB/TCRTOBJY
_ YYCMD  Command                   *CMD           N     400  &PRDLIB/TCRTOBJY
_ YYDSPF AS/400 Display File       *FILE          N     450  &PRDLIB/TCRTOBJY
_ YYLF   AS/400 Logical File       *FILE          Y     425  &PRDLIB/TCRTOBJY
_ YYPF   AS/400 Physical File      *FILE          Y     410  &PRDLIB/TCRTOBJY
                                                                      More...
F3=Exit       F6=Add Type       F8=User-defined parms          F24=More Keys
```

Press **Enter**.

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



```
 6/15/01                     Change a Type Code              Your Company, Inc.
16:05:04                                                     YOURSYS

Type code  . . . . . UTBIO                         CM scheme  . . . . . UTBIO
Description  . . . . Employee Biography            Attribute  . . . . .
Object type  . . . . *UTBIO                        Source file name . .
Sequence . . . . . . 100                           Data object  . . . . N
Create command . . . __________________________________________________________
_______________________________________________________________________________

_______________________________________________________________________________
                                       "&RF" = Reference      "&X0"-"&X9" = Exit
"&OB" = Object name                    "&TO" = Test object name
"&LI" = Library name                   "&TL" = Test object library name
"&TY" = Type code                      "&TR" = Target release
"&SM" = Source member name             "&FM" = From source member name
"&SF" = Source file name               "&FF" = From source file name
"&SL" = Source library name            "&FL" = From source library name
"&U0" = Generation options             "&U1" = Debugging views
"&U2" = Optimization level             "&U3" = Compile options
"&U4" = Source listing options         "&U5" = Unassigned
"&U6" = Unassigned                     "&U7" = Unassigned
"&U8" = Unassigned                     "&U9" = Unassigned
F3=Exit  F4=Prompt/select   F7=Applications  F12=Cancel  F22=Long command
```

8. Return to TURNOVER® for iSeries v100 Main Menu option **1** (***Work with Application Definitions***) and type option **14** (***Create missing items***) next to the UT application. Unless it has been created previously, you will be prompted to create library UTBIODEV (development library). Press **Enter** to create the library.

9. Now you are ready to create a Programmer Worklist and start experimenting with the UTBIO CM scheme. To start with, you can work with two existing employee biographies:

- Department MARKETING, employee COUNTERB
- Department SALES, employee KRAAPS.

UTBIO scheme objects

In addition to the files restored during the restore operation (described previously), these objects are restored when you restore the UTBIO CM scheme:

- UTBIO (CLLE) – This is the method program for all the scheme's overridden CM methods. These include ***CHKOBJ**, ***CRTDUPOBJ**, ***DLTOBJ**, ***MOVOBJ**, and ***RTVOBJD**.
- UTBIOR (RPGLE) – Called by UTBIO to perform the I/O on the employee biography and alias files.
- UTSLTBIO (CMD, CLLE) – The Select Employee Biography command and command processing program. (See ***Creating a selection program for user-defined objects*** on page 54.)
- UTBIOSLT (DSPF, RPGLE) – The display file and program (called by UTSLTBIO) used to add employee biographies to the Programmer Worklist from a selection list, generating aliases when necessary. (See ***Creating a selection program for user-defined objects*** on page 54.)

The source code for all the UTBIO objects is stored in these source files in the TURNOVER® for iSeries v100 product library:

- ADDSSRC (PFs, LF, DSPF)
- ACMDSRC (CMD)
- ACLSRC (CLLE)
- ARPGLESRC (RPGLE).

Review these source members, paying special attention to the remaining topics for this UTBIO CM scheme.

Overriding CM methods to *NONE

The UTBIO CM scheme uses ***NONE** as its the default method program. Actually, ***NONE** is appropriate for most of the methods for this particular scheme because they require no action by TURNOVER® for iSeries v100 but we still want some reflection of a "positive" completion. The exceptions to this are the ***CHKOBJ**, ***CRTDUPOBJ**, ***DLTOBJ**, ***MOVOBJ**, and ***RTVOBJD** methods, which have all been overridden to use program UTBIO as the method program; and the ***CHKSRC** method, which has been overridden to use ***TURNOVER**.

CHKSRC** uses ***TURNOVER** rather than ***NONE**. Using ***NONE** for the ***CHKSRC** method would cause Programmer Worklists to appear as though source exists at all levels, even though the UTBIO scheme is used to manage a non-source object type (that is, no **"&SM"**, **"&SF"**, **"&SL"** variables in global create command, which is blank). By specifying ***TURNOVER** instead, the Programmer Worklists will always show **N** in the ***Src column (because TURNOVER® for iSeries v100's method of checking for source is to automatically return **N** for non-source object types).

Whether you should use ***TURNOVER** or ***NONE** for a scheme's method is subtle, depending on the nature of the object type you are managing and the method you are overriding. When creating your own CM schemes, you might need to experiment some to determine which of these methods is appropriate for which methods (and which methods need to be overridden to programs).

Using object aliases in CM schemes

Because TURNOVER® for iSeries v100's database does not support object names longer than 10 characters, it is sometimes necessary to devise an aliasing scheme to manage user-defined object types that require more than 10 characters for unique identification. The UTBIO CM scheme is an example of this. File UTBIO (Employee Biography File) is keyed by Department Code (length is 10; data type is Character) and Employee (length is 10; data type is Character), which is a total key length of 20 characters. This scheme uses an alias file (UTBIOALIAS) to cross-reference 10-character aliases with the actual department codes and employee IDs that the aliases represent. Program UTBIO (and UTBIOR, which is called by UTBIO), the method program for the ***CHKOBJ**, ***CRTDUPOBJ**, ***DLTOBJ**, ***MOVOBJ**, and ***RTVOBJD** methods, is responsible for translating aliases to department codes and employee IDs to perform the requested CM method.

In file UTBIO, the format of the Employee ID field is last name, first initial (Bill Smith = SMITHB) and each department (MARKETING, SALES) starts with a unique letter. A logical aliasing scheme, which this sample uses, is to generate aliases in this format:

First letter of department + First letter of Employee ID + Last letter of Employee ID + *nnn*

where *nnn* is 001, 002 (if there is already an alias with the same first three characters), 003, and so forth. For example, the alias for Bill Smith in Marketing would be MSB001 (if there is not already an alias starting with MSB).

Of course, this aliasing scheme is unique to this example. When creating your own CM schemes, you will have to come up with your own aliasing schemes, using whatever formula is appropriate for your object type.

The UTBIO scheme requires an alias file (UTBIOALIAS) and logical file (UTBIOALILF) to keep track of aliases. However, in some cases it might be possible to use an aliasing scheme that does not require such a file.

For example, suppose you were creating a CM scheme that manages records in a file that is keyed by Category (length is 10; data type is Character) and Part Code (length is 8; data type is Character). The total length of the key exceeds TURNOVER® for iSeries v100's limit of 10 for object names. But, suppose you can safely assume that no two categories will begin with the same letter. If that were the case, you could use a scheme where the object alias has the format C*pppppppp* (where C is the first letter of the Category, and *pppppppp* is the entire part code). Because the alias is not longer than 10 characters, not special alias file is needed.

Creating a selection program for user-defined objects

As long as the CM scheme is defined appropriately, user-defined object types are fully supported by the Programmer Worklist. However, for user-defined object types, we do not have the convenience of pressing **F20** and selecting (using TURNOVER® for iSeries v100's user-defined PDM option) objects to add to the worklist from a list of objects or members. Of course, you can press **F6** and type the object names and type codes to manually add the objects to the worklist. But this could quickly become tedious. And, if your scheme requires an alias file, you will also need to add the necessary records to that file before you can add the object to the worklist (using an alias name).

To simplify this, you can write a command that brings up a selection list of your objects and adds selected objects to the active worklist, generating aliases as necessary. This program could then be called from the worklist command line (or by a user-defined worklist option if there is already at least one object on the worklist against which to execute the option).

Such a program would be somewhat complex. However, the UTBIO scheme includes such a command (UTSLTBIO) that you can use as is or modify to suit your needs when creating similar commands for your own CM schemes.

Carefully review the source code for these objects:

- UTSLTBIO (CMD, CLLE)
- UTBIOSLT (DSPF, RPGLE).

There are extensive comments in the source code indicating the points where the source would require modifications to accommodate your own CM scheme.

SAMPLE 3 – UTDDG SCHEME (JDEDWARDS DATA DICTIONARY GLOSSARY TEXT)

The **UTDDG** CM scheme is a sample scheme that manages changes to JDEdwards Data Dictionary Glossary text (referred to in this section as *Data Dictionary Glossary Object*). The main driver file is the JDEdwards F9203 Data Item Alpha Descriptions file. The file and its fields are:

F9203 – Data Item Alpha Descriptions File

Field	Description	Length	Data Type
FRDTAI	Data Item (Key)	10	Character
FRLNGP	Language Group (Key)	2	Character
FRSYR	Reporting System (Key)	4	Character
FRDSCA	Description Alpha	40	Character
FRDSCC	Description Compressed	40	Character
FRSCRN	Screen (Key)	10	Character

Because the total length of the file's key is 26 (see lengths and data types above for Data Item, Language Group, Reporting System, and Screen), you cannot identify a Data Dictionary Glossary object using a 10-character object name such as TURNOVER® for iSeries v100's database supports. Consequently, this scheme uses aliases, which are stored in an alias file. (For additional details, see ***Using object aliases in CM schemes*** on page 61.) The alias file and its fields are:

UTDDGALIAS – Data Dictionary Glossary Alias File

Field	Description	Length	Data Type
ALIAS	Alias (Key)	10	Character
DTAI	Data Item	10	Character
LNGP	Language	2	Character
SYR	System Code	4	Character
SCRN	Screen	10	Character

For retrieving an existing alias for a Data Dictionary Glossary object, there is also a logical file, UTDDGALILF, over the UTDDGALIAS file. The key fields for the UTDDGALILF file are Data Item, Language, System Code, and Screen.

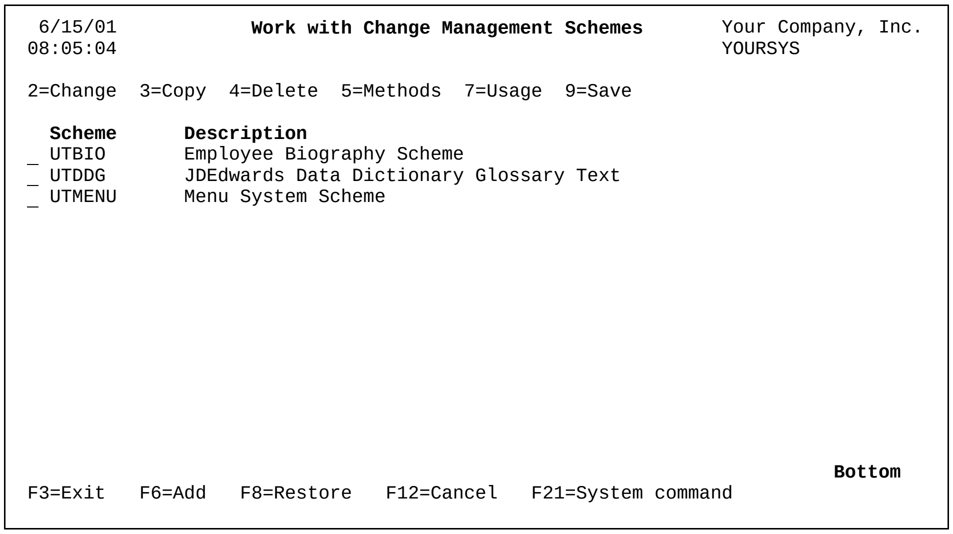
Restoring the UTDDG scheme

Restoring the UTDDG scheme is slightly different from restoring the UTBIO or UTMENU schemes. Restoring the UTDDG scheme restores the UTDDG library with the method programs, and also creates a TURNOVER® for iSeries v100 CM scheme. The CM scheme will not be assigned automatically to any existing TURNOVER® for iSeries v100 applications, nor will it be assigned automatically to the UT sample application.

To restore the UTDDG scheme, you must perform these steps:

1) Sign on as a user profile with authority to restore libraries and TURNOVER® for iSeries v100 application definitions to the system.

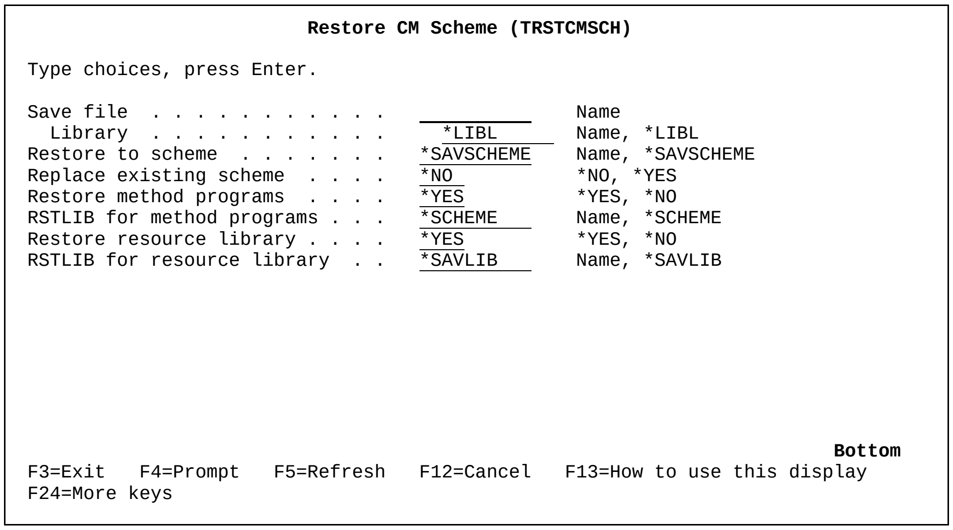
2) On a TURNOVER® for iSeries v100 command line, type **TWRKCMSCH** and press **Enter**. The ***Work with Change Management Schemes*** panel appears:



```
 6/15/01            Work with Change Management Schemes       Your Company, Inc.
08:05:04                                                      YOURSYS

2=Change  3=Copy  4=Delete  5=Methods  7=Usage  9=Save

  Scheme      Description
_ UTBIO       Employee Biography Scheme
_ UTDDG       JDEdwards Data Dictionary Glossary Text
_ UTMENU      Menu System Scheme










                                                                        Bottom
F3=Exit   F6=Add   F8=Restore   F12=Cancel   F21=System command
```

3) If the UTDDG scheme is already listed on this panel, proceed to step 6.

4) Press **F8** (**Restore**). The ***Restore CM Scheme (TRSRCMSCH)*** panel appears:



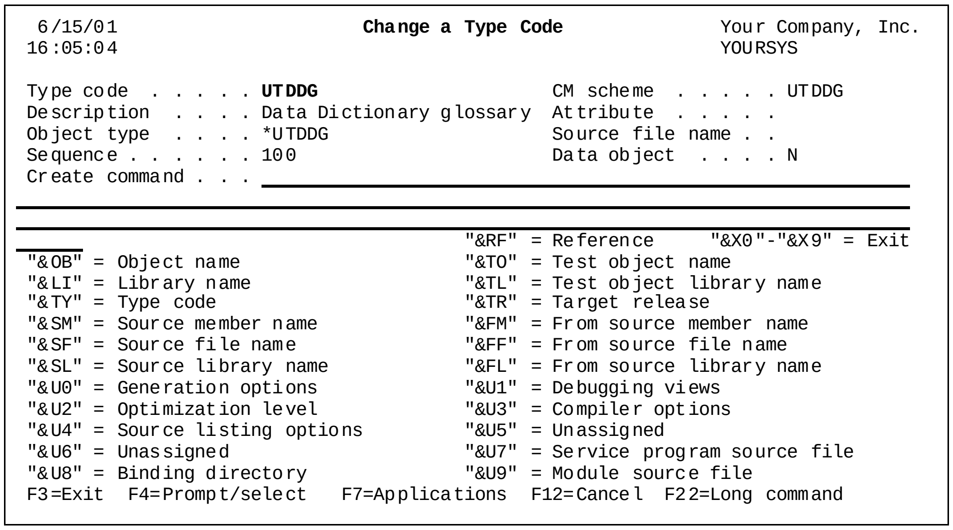
```
                         Restore CM Scheme (TRSTCMSCH)

Type choices, press Enter.

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








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

On this panel, type **UTDDGSAVF** for the save file name. Accept the default values for the other parameters (the save file is in the TURNOVER® for iSeries v100 product library). Press **Enter**. TURNOVER® for iSeries v100 then restores these items:

a) The UTDDG scheme definition.

b) Library UTDDG (the resource library), which contains the Data Dictionary Glossary Alias file (described on page 56), the UTDDG CM method programs, and the source to the objects being restored.

5) Select TURNOVER® for iSeries v100 Main Menu option **8** (***Utility Menu***) and then option **4** (***Work with type code definitions***). If the UTDDG type code is not listed on the ***Work with Type Codes*** panel (you might have to page down to see it), you need to create it. If the type code is already there, then type **2** (and press **Enter**) to view it. In either case, you must make sure that the type code definition matches this:[5]



```
 6/15/01                      Change a Type Code              Your Company, Inc.
16:05:04                                                      YOURSYS

Type code  . . . . . UTDDG                        CM scheme  . . . . . UTDDG
Description  . . . . Data Dictionary glossary   Attribute  . . . . .
Object type  . . . . *UTDDG                       Source file name . .
Sequence . . . . . . 100                          Data object  . . . . N
Create command . . . 


                                        "&RF" = Reference      "&X0"-"&X9" = Exit
"&OB" = Object name                     "&TO" = Test object name
"&LI" = Library name                    "&TL" = Test object library name
"&TY" = Type code                       "&TR" = Target release
"&SM" = Source member name              "&FM" = From source member name
"&SF" = Source file name                "&FF" = From source file name
"&SL" = Source library name             "&FL" = From source library name
"&U0" = Generation options              "&U1" = Debugging views
"&U2" = Optimization level              "&U3" = Compiler options
"&U4" = Source listing options          "&U5" = Unassigned
"&U6" = Unassigned                      "&U7" = Service program source file
"&U8" = Binding directory               "&U9" = Module source file
F3=Exit  F4=Prompt/select   F7=Applications  F12=Cancel  F22=Long command
```

6) Now you need to assign the UTDDG type code to a TURNOVER® for iSeries v100 application that is used to manage JDEdwards objects. Select TURNOVER® for iSeries v100 Main Menu option **1** (***Work with Application Definitions***). On the ***Work with Application Definitions*** panel, type **2** (***Change***) next to a valid JDEdwards application, then type option **14** (***Type codes***) next to any level. On the ***Type Codes*** panel, press **F4** (***Add/Delete types***), page down until you see the UTDDG type code, and type **1** next to it. Press **Enter**. When you are prompted to do so, be sure to include this type code for any additional levels of the application.

7) Now you are ready to create a Programmer Worklist and start experimenting with the UTDDG CM scheme.

[5] Make the necessary changes to have your UTDDG type code what is shown here.

UTDDG scheme objects

In addition to the files restored during the restore operation (described previously), these objects are restored when you restore the UTDDG CM scheme:

- UTDDG (CLLE) – This is the method program for all the scheme's overridden CM methods. These include ***BROWSE**, ***CHKOBJ**, ***CRTDUPOBJ**, ***DLTOBJ**, ***EDTSRC**, ***MOVOBJ**, and ***RTVOBJD**.
- UTDDGR (RPGLE) – Called by UTDDG to perform the I/O on the Data Dictionary Glossary alias file, and to call the JD Edwards routines to transfer Data Dictionary control records (P996301).
- UTDDGC1 (CLLE) – Uses TURNOVER® for iSeries v100 routines to check if the base Data Dictionary object exists in a library. The base object must exist before a glossary variant can be created in that library.
- UTDDGSLT (CMD, CLLE, DSPF) – The Select Data Dictionary Glossary object command and command processing program. (See ***Creating a selection program for user-defined objects*** on page 62.)
- UTDDGSLTR (RPGLE) – The display file and program (called by UTSLTDDG) used to add Data Dictionary Glossary objects to the Programmer Worklist from a selection list, generating aliases when necessary. (See ***Creating a selection program for user-defined objects*** on page 62.)
- UTDDGEDT (CMD, CLLE, DSPF) – Sets up the environment to edit a specified Data Dictionary Glossary object, then calls the JDEdwards program P92001 to present the edit screen. (**Note:** When editing a specific screen, you must first position to that screen before starting to edit.)
- UTDDGVIEW (CMD, CLLE, DSPF) – Can be specified in a user-defined option to be executed next to a particular worklist item. This presents a window that translates the alias name to the actual Data Dictionary Glossary object.
- UTDDGVIEWR (RPGLE) – Called by UTDDGVIEW.

The source code for all the UTDDG objects is stored in these source files in the resource library UTDDG:

- QDDSSRC (PFs, LF, DSPF)
- QCMDSRC (CMD)
- QCLSRC (CLLE)
- QRPGLESRC (RPGLE).

Review these source members, paying special attention to the remaining topics for this UTDDG CM scheme.

To use these programs in your environment, you must make sure that the UTDDG library is included in your library list either by adding UTDDG to your TURNOVER® for iSeries v100 environment record, or by using the **ADDLIBLE** command.

Overriding CM methods to *NONE

The UTDDG CM scheme uses ***NONE** as its default method program. Actually, ***NONE** is appropriate for most of the methods for this scheme. The exceptions to this are the ***BROWSE**, ***CHKOBJ**, ***CRTDUPOBJ**, ***DLTOBJ**, ***EDTSRC**, ***MOVOBJ**, and ***RTVOBJD** methods, which have all been overridden to use program UTDDG as the method program; and the ***CHKSRC** method, which has been overridden to use ***TURNOVER**. For additional explanation, see ***Overriding CM methods to *NONE*** on page 53. Except for the scheme name (which in that case is UTBIO), the text in the second paragraph of that section also relates to this UTDDG scheme.

Using object aliases in CM schemes

Because TURNOVER® for iSeries v100's database does not support object names longer than 10 characters, it is sometimes necessary to devise an aliasing scheme to manage user-defined object types that require more than 10 characters for unique identification. The UTDDG CM scheme is an example of this. File F9203 (Data Item Alpha Descriptions) is keyed by Data Item (length is 10; data type is Character), Language (length is 2; data type is Character), Reporting System (length is 4; data type is Character), and Screen (length is 10; data type is Character), which is a total key length of 26 characters. This scheme uses an alias file (UTDDGALIAS) to cross-reference 10-character aliases with the actual F9203 records that the aliases represent. Program UTDDG (and UTDDGR, which is called by UTDDG), the method program for the ***BROWSE**, ***CHKOBJ**, ***CRTDUPOBJ**, ***DLTOBJ**, ***EDTSRC**, ***MOVOBJ**, and ***RTVOBJD** methods, is responsible for translating aliases to the corresponding F9203 records to perform the requested CM method.

In file F9203, most Data Items consist of 4 positions (even though 10 are available), and the system reporting code must be blank if a screen is being defined. A logical aliasing scheme, which this sample uses, is to generate aliases in the format DDDD_L_999:

> First 4 positions of the Data Item + an underscore + first 1 position of the Language code + an underscore + *nnn*

where *nnn* is 001, 002 (if there is already an alias with the same first three characters), 003, and so forth.

For example, the alias for Data Item ACL1 for language French, screen V12010 would be ACL1_F_001 (if there is not already an alias starting with 'ACL1_F_'). If the Data item was AD, base language, Screen V069951, underscores would be used to fill the blanks, and the alias AD_____001 would be used (5 underscores in this example).

You might want to use this aliasing scheme, or create your own that incorporates part of the screen ID. Use whatever scheme works best for your setup. If you change the aliasing scheme, be sure to review all the related programs in the UTDDG scheme.

Creating a selection program for user-defined objects

As long as the CM scheme is defined appropriately, user-defined object types are fully supported by the Programmer Worklist. However, for user-defined object types, we do not have the convenience of pressing **F20** and selecting (using TURNOVER® for iSeries v100's user-defined PDM option) objects to add to the worklist from a list of objects or members. Of course, you can press **F6** and type the object names and type codes to manually add the objects to the worklist. But this could quickly become tedious. And, if your scheme requires an alias file, you will need to also add the necessary records to that file before you can add the object to the worklist (using an alias name).

To simplify this, you can write a command that brings up a selection list of available Data Dictionary Glossary objects and adds selected objects to the active worklist, generating aliases as necessary. This program could then be called from the worklist command line (or by a user-defined worklist option if there is already at least one object on the worklist against which to execute the option).

The UTDDG scheme includes a command (UTSLTDDG) that you can use as is or modify to suit your needs.

Carefully review the source code for these objects:

- UTDDGSLT (CMD, CLLE, DSPF)
- UTDDGSLTR (RPGLE).

The UTDDGSLT command hides the system reporting code. It does this because it is not valid to define an override for a screen along with a specific system reporting code.

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.