

TURNOVER[®] Lifecycle Manager

JD Edwards World interface

Publication information

TN4J-2100-00 (January 2021)

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

Copyright notice

TURNOVER® Lifecycle Manager (the Programs and associated materials) is a proprietary product of SoftLanding Systems, Inc. – a division of UNICOM Global. The Programs have been provided pursuant to License Agreement containing restrictions on their use. The programs and associated materials contain valuable trade secrets and proprietary information of SoftLanding Systems, Inc. and are protected by United States Federal and non-United States copyright laws. The Programs and associated materials may not be reproduced, copied, changed, stored, disclosed to third parties, and distributed in any form or media (including but not limited to copies on magnetic media) without the express prior written permission of SoftLanding Systems, Inc., 1 Vose Farm Road, Suite 110, Peterborough, NH 03458, USA.

TURNOVER® Lifecycle Manager (the Programs and associated materials) is a proprietary product of UNICOM Systems, Inc. – a division of UNICOM Global. The Programs have been provided pursuant to License Agreement containing restrictions on their use. The programs and associated materials contain valuable trade secrets and proprietary information of UNICOM Systems, Inc. and are protected by United States Federal and non-United States copyright laws. The Programs and associated materials may not be reproduced, copied, changed, stored, disclosed to third parties, and distributed in any form or media (including but not limited to copies on magnetic media) without the express prior written permission of UNICOM Systems, Inc., UNICOM Plaza Suite 310, 15535 San Fernando Mission Blvd., Mission Hills, CA 91345 USA.

TURNOVER® Lifecycle Manager

© Copyright 2005-2021 All Rights Reserved. Softlanding Inc. – a division of UNICOM Global.

No part of this Program may be reproduced in any form or by electronic means, including the use of information storage and retrieval systems, without the express prior written consent and authorization of SoftLanding Systems, Inc.

No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, without prior written permission from SoftLanding Systems, Inc.

© Copyright 2005-2021 All Rights Reserved. UNICOM Systems, Inc. – a division of UNICOM Global.

No part of this Program may be reproduced in any form or by electronic means, including the use of information storage and retrieval systems, without the express prior written consent and authorization of UNICOM Systems, Inc.

No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, without prior written permission from UNICOM Systems, Inc.

Disclaimer

We cannot guarantee freedom from, or assume any responsibility or liability for technical inaccuracies or typographical errors. The information herein is furnished for informational use only and should not be construed as a commitment by SoftLanding Systems, Inc. – a division of UNICOM Global.

Trademarks

The following are trademarks or registered trademarks of Macro 4 Limited, SoftLanding Systems, Inc. and/or of UNICOM Systems, Inc. in the United States and/or other jurisdictions worldwide: UNICOM, UNICOM Systems, Macro 4, SoftLanding, AUTOMON, CENTRAL, EXPRESSDESK, SUPERMON, TURNOVER.

For a list of third-party products, companies, trademarks, and registered trademarks that might be referenced in this manual, see www.softlanding.com/trademarks.

Contents

1	Using TURNOVER® Lifecycle Manager with JD Edwards World	5
	Using the TURNOVER® Lifecycle Manager/JD Edwards World interface	5
	Unique JD Edwards World objects	6
	TURNOVER® Lifecycle Manager type codes for unique JD Edwards World objects	6
	TURNOVER® Lifecycle Manager application definitions for JD Edwards World applications	9
	Configuring TURNOVER® Lifecycle Manager to manage JD Edwards World objects	11
	Managing multiple JD Edwards World application releases	12
2	Setting up TURNOVER® Lifecycle Manager for World applications	13
	Setting up the interface using the setup wizard	13
	Setting up the interface manually	26
	Changing the remote TURCRARE job descriptions	37
	Reviewing the JD Edwards World application type codes	38
	Setup wizard: TSETUPJDE command parameters	41
3	Managing JD Edwards World components	42
	Managing JD Edwards World native object types	42
	Creating JD Edwards World programs with embedded SQL	46
	Creating additional JD Edwards World object types	47
	Specifying the target release	48
	Retrieving compile options from the SVR	49
	Using an exit point to update the SVR	50
4	Managing Dream Writer, World Writer, and FASTR versions	52
	Setting up versions	52
	Using version support	53
	More about JD Edwards World vocabulary overrides (JDEVO)	56
5	JD Edwards World data-driven object types	59
	Copy JDE Object command (CPYJDEOBJ)	60
	Exploding JD Edwards World data-driven object types	61
	Language, system, and version support (for JDEDD, JDEDW, JDEWWV, JDEFSV, and JDEUDC)	62
	Language support for JD Edwards World soft-coded types	64

6	Developing JD Edwards World objects with TURNOVER® Lifecycle Manager	67
	Managing JD Edwards World objects from a programmer worklist	68
	TURNOVER® Lifecycle Manager commands for JD Edwards World applications	72
	Menu language preferences	75
	Recording SAR numbers in the SVR detail record	76
7	DBCS versions of JD Edwards software	77

1 Using TURNOVER[®] Lifecycle Manager with JD Edwards World

You can use TURNOVER[®] Lifecycle Manager to manage daily changes to the Oracle's JD Edwards World.

Features

- All TURNOVER[®] Lifecycle Manager features, such as checkout, check-in, promotion, and program development using TURNOVER[®] Lifecycle Manager's Programmer Worklist Manager, support JD Edwards World objects seamlessly.
- You can access JD Edwards World SVR functions, such as a command line, repository services, where-used cross-reference, and so on, from the worklist.
- The TURNOVER[®] Lifecycle Manager cross-reference system displays the effect of any proposed change on all JD Edwards World IBM i objects.
- TURNOVER[®] Lifecycle Manager's Synchronizer system enables you to merge a new release of a JD Edwards World application with your changes. It includes a source compare and merge utility.

Using the TURNOVER[®] Lifecycle Manager/JD Edwards World interface

Installing the interface

No special installation is required to use the interface, unless you are managing a DBCS version of the JD Edwards software: see *'DBCS versions of JD Edwards software' on page 77*.

Configuring the interface

However, you must configure some components of TURNOVER[®] Lifecycle Manager to ensure that objects unique to JD Edwards World applications are handled properly: see *'Setting up TURNOVER[®] Lifecycle Manager for World applications' on page 13*.

Support notes

- TURNOVER[®] Lifecycle Manager supports JD Edwards World through Release 9.41.

- You can check out standard IBM i objects using the Eclipse-based TURNOVER® Lifecycle Manager plugins for RDi; however, Oracle does not provide an Eclipse-based user interface for JDE softcoded types. Therefore, support for some JDE types is limited when you are using the Eclipse-based TURNOVER® Lifecycle Manager user interface to manage your JDE software.

Unique JD Edwards World objects

JD Edwards World applications use a repository file, called the *software version repository* (SVR). This repository controls compile time parameters for JD Edwards World IBM i objects, such as physical files, logical files, printer files, programs, and so on. There are special programs supplied by JD Edwards World that create these objects. These programs perform processing before running the actual IBM i compile commands to create JD Edwards World IBM i objects.

JD Edwards World applications also use some “data-driven” object types. These include Data Dictionary items, DREAM Writers, FASTRs, menus, processing options, user-defined codes, Vocabulary Overrides, and World Writers.

JD Edwards World data-driven objects are not really objects; instead, they consist of clusters of related records that are in several different JD Edwards World definition files. From a change management perspective, you can think of these record clusters as objects. Each cluster can be checked out, modified, promoted, and distributed to production computers, just as any other IBM i object can be when TURNOVER® Lifecycle Manager is being used.

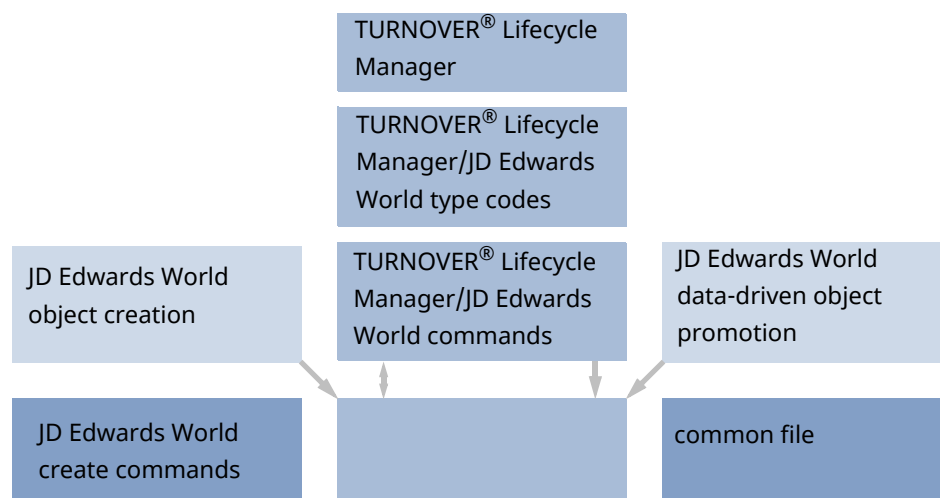
To maintain JD Edwards World applications with TURNOVER® Lifecycle Manager, you set up TURNOVER® Lifecycle Manager application definitions, in which you identify which JD Edwards World objects you want to maintain, by using TURNOVER® Lifecycle Manager type codes.

TURNOVER® Lifecycle Manager type codes for unique JD Edwards World objects

TURNOVER® Lifecycle Manager handles the JD Edwards World objects with several type code definitions. It also requires special create commands that are specific to JD Edwards World objects. The type codes definitions tell TURNOVER® Lifecycle Manager which special create command to use when promoting the JD Edwards World application object from one level of your World application to another.

The TURNOVER® Lifecycle Manager type code definitions are already in the type code file, and the create commands are in the product library.

TURNOVER® Lifecycle Manager as it relates to JD Edwards World applications



JD Edwards World IBM i object type codes and create commands

For each JD Edwards World IBM i object type, there is a TURNOVER® Lifecycle Manager object create command.

Type code	Create command
CLPJDE	JDECLP
DSPJDE	JDEDSPF
LFJDE	JDELF
PFJDE	JDEPF
PRTJDE	JDEPRTF
RPGJDE	JDERPG

Any of these can be specified for a TURNOVER® Lifecycle Manager application definition, along with any other types that may be appropriate. When you associate one of these type codes with an application definition, you can check out, check in, develop, and promote your JD Edwards World objects just as you would other object types.

Each create command calls a program supplied by JD Edwards World that gets the necessary information from the JD Edwards World repository, and then compiles the programs, by using standard IBM i compiler commands.

For more information, see:

'Type code specifications for JD Edwards World IBM i objects' on page 33

'Reviewing the JD Edwards World application type codes' on page 38.

Data-driven object type codes and create commands

TURNOVER® Lifecycle Manager includes the following data-driven object type codes.

The JD Edwards World data-driven object types all use the same create command: `CPYJDEOBJ`. The values for the `CPYJDEOBJ` parameter Object type enable it to handle the appropriate data-driven object.

TURNOVER® Lifecycle Manager type code	Description	CPYJDEOBJ object type parameter
JDEDD	Data Dictionary item	*JDEDD, *JDD
JDEDW	DREAM Writer	*JDEDW, *JDW
JDEDWV	Dream Writer Versions	*JDEDWV
JDEFSV	FASTR Versions	*JDEFSV
JDEMNU	Menus	*JDEMNU
JDEPO	Processing options	*JDEPO
JDESV	Software Version Repository (SVR) items	*JDESV
JDEUDC	User-defined codes	*JDEUDC, *JUD
JDEVO	Vocabulary override	*JDEVO, *JVO
JDEWWV	World Writer Versions	*JDEWWV

For more information about the data-driven object type codes, and examples of each, see:

'Type code specifications for JD Edwards World data-driven objects' on page 34

'Reviewing the JD Edwards World application type codes' on page 38.

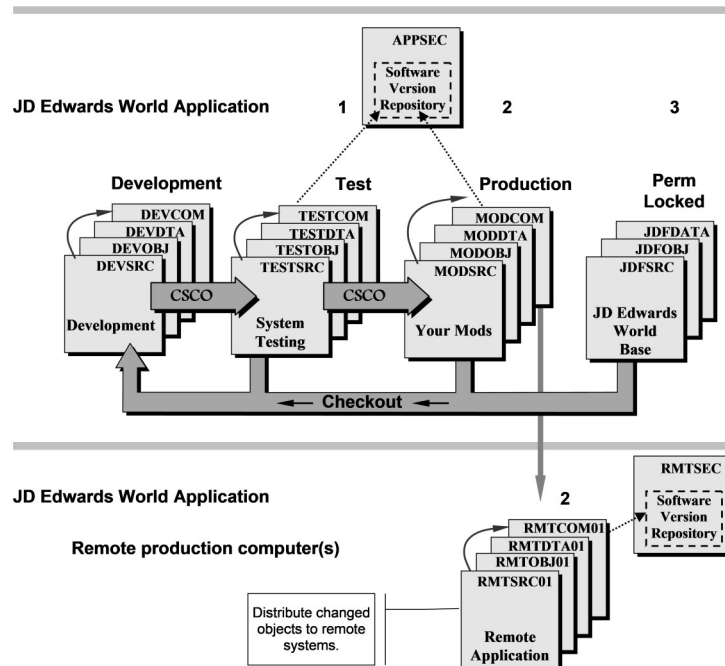
Managing DDG text and other unique JD Edwards World objects

TURNOVER® Lifecycle Manager manages all the major JD Edwards World soft-coded types.

To check out and promote other types, such as Data Dictionary Glossary (DDG) text, you can create your own user-defined object types. For more information, see "Managing User-Defined Object Types" in the *TURNOVER® Lifecycle Manager Administrator's Guide*.

TURNOVER® Lifecycle Manager application definitions for JD Edwards World applications

This diagram shows a typical JD Edwards World application, suitable for managing day-to-day changes to World applications. It uses four levels; if you use more or less levels, the general pattern of object movement is the same.



In this example:

- Program development occurs on one computer, and objects or source members (or both) are distributed to another computer.
- Testing occurs on the development computer.
- Changes are promoted to a production level (2) and then distributed to several production computers, each one set up the same way.

Change activity

The JD Edwards World base libraries are defined at the highest application level. This level is "permanently locked," meaning that no changes are made to the base source or objects except when you apply either changes or a new JD Edwards World application release. Change activity for your JD Edwards World application might follow this pattern:

- 1 Source for programs and other objects is checked out to a programmer's development library (or a common development or project library). TURNOVER® Lifecycle Manager looks first in MODSRC and, if not found there, then in JDFSRC, and then copies it to development library. New object names are reserved and recorded in the TURNOVER® Lifecycle Manager checkout file.

A naming convention, as when object names must start with JDE or XXX, can be enforced by application and object type.

- 2 The programmer changes the source, compiles it, and then tests it in the designated development library.
- 3 The programmer builds a TURNOVER® Lifecycle Manager form to promote to Level 1, and run an error-check.
- 4 The Level 1 form is approved and run.

TURNOVER® Lifecycle Manager copies source into TESTSRC library and compiles objects into the test libraries for level 1.

The library list for the level 1 form is:

QTEMP,TESTSRC, TESTOBJ, TESTDTA, TESTCOM, MODSRC, MODOBJ, MODDTA, MODCOM, APPSEC,
JDFSRC, JDFOBJ, JDFDATA, QGPL

The programmer's development library should *not* be in the library list.

The Test libraries contain only objects being changed.

- 5 The administrator copies one or more Level 1 forms to Level 2 (preparing them to run for Level 2).
- 6 The Level 2 form is approved and run. TURNOVER® Lifecycle Manager copies source into MODSRC, and then compiles objects into the MOD level libraries. The Existing data is copied to new files. LFs are recreated over PFs.

The library list for Level 2 run is:

QTEMP, MODSRC, MODOBJ, MODDTA, MODCOM, APPSEC, JDFSRC, JDFOBJ, JDFDATA, QGPL

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

- 7 After the objects have been successfully promoted to the mod level (production), they are distributed to one or more production computers, and installed there.

Creating application definition models

Application definition models 8, 9, 20JD, and 21JD are examples of definitions for managing vendor-supplied software. See *'Definition models for JD Edwards World applications' on page 10*.

You can also use the JD Edwards World Setup Wizard to create your JD Edwards World application definitions: see *'Setting up the interface using the setup wizard' on page 13*.

Definition models for JD Edwards World applications

You can create your own JD Edwards World applications by copying and changing the application definition models 20JD and 21JD which are included with TURNOVER® Lifecycle Manager.

- 1 Select TURNOVER® Lifecycle Manager main menu option 3, and then authorize yourself to these definitions.
- 2 Select TURNOVER® Lifecycle Manager main menu option 1, and then view the application definition models as you review this material.

The model application definitions already refer to the JD Edwards World type codes (see *'JD Edwards World IBM i object type codes and create commands' on page 7*). You can use these application definitions even if there are more levels than you need: just copy the application levels you want from one of the models, and then set the library names and other information appropriately for each level. If you need to change any of the other application defaults, see "Working with application definitions" in the *TURNOVER® Lifecycle Manager User Guide*.

Each application level is defined by its own application definition. For each level, you need to supply information that is specific to JD Edwards World applications for the following defaults:

- application default owner
- target libraries
- from libraries
- type codes.

The TURNOVER® Lifecycle Manager global type code file already contains codes for the JD Edwards World object types. When you define a JD Edwards World application, you can include in its default type code list the World object types that are appropriate to your application, and remove those you do not need. You can also change the type codes as needed: see *'TURNOVER® Lifecycle Manager type codes for unique JD Edwards World objects' on page 6*.

You might want to consider changes to other application parameters, such as:

- object name masks
- create parms
- authority.

Configuring TURNOVER® Lifecycle Manager to manage JD Edwards World objects

- 1 Create an application schematic.

See *'TURNOVER® Lifecycle Manager application definitions for JD Edwards World applications' on page 9*.

- 2 Read *'Using the TURNOVER® Lifecycle Manager/JD Edwards World interface' on page 5*.
- 3 Make sure that you should know the names of your JD Edwards World program, data, common, and security libraries.

4 To configure TURNOVER® Lifecycle Manager to handle JD Edwards World objects, you can either:

- run the setup wizard
- configure TURNOVER® Lifecycle Manager manually.

If you change the application structure (for example, by adding or removing levels), change application libraries, or update your JD Edwards release, use the setup wizard after your changes to verify the new setup.

Managing multiple JD Edwards World application releases

To support multiple releases or installations of JD Edwards World applications simultaneously, you can use either the traditional environment setup or the alternative environment setup.

Traditional environment setup

This configuration is useful if you have separate teams working in each JDE environment, or if you do not plan to switch between environments often.

- 1 Sign on as QSECOFR.
- 2 Use the `WRKTOENV` (Work with TURNOVER® Lifecycle Manager Environment) command to create a separate TURNOVER® Lifecycle Manager environment for each JD Edwards World application release you need to support.
- 3 In each TURNOVER® Lifecycle Manager environment you created, associate TURNOVER® Lifecycle Manager with the appropriate programs and files for the intended JD Edwards World application release.

To do this, do one of the following:

- run the Setup wizard: see *'Setting up the interface using the setup wizard' on page 13*
- configure TURNOVER® Lifecycle Manager manually: see *'Setting up the interface manually' on page 26*.

Alternative environment setup

This configuration is useful if the same team makes many changes in each JDE environment.

To configure this type of environment, see *'Setting up the interface using the setup wizard' on page 13*.

2 Setting up TURNOVER® Lifecycle Manager for World applications

You can set up TURNOVER® Lifecycle Manager for World applications either:

- by using the setup wizard: see *'Setting up the interface using the setup wizard' on page 13*
- manually: see *'Setting up the interface manually' on page 26.*

Setting up the interface using the setup wizard

- 1 Sign on as QSECOFR.
- 2 On the TURNOVER® Lifecycle Manager main menu, press F21 to display a command line.
- 3 Type TSETUPJDE, and then press Enter.

The Set Up JDE Interface panel appears.

```

Set Up JDE Interface

This command can be run at any time, either to set up the TURNOVER® to JDE
interface for the first time, or to identify and correct potential problems
with the existing setup. Select the setup options that you want to check,
add press Enter.

1=Select

  1 Overview
  1 Check TURNOVER® environment
  1 Check Programmer Worklist user-defined options
  1 Check JDE application definitions
  1 Check application library lists
  1 Check type code definitions
  1 Check for JDE data driven type files in application libraries
  1 Check SVR automatic update
  1 Check Jobd for TURCRARE on remote systems
  1 Enable Alternate Environments

F3=Exit   F12=Cancel   F21=System command

```

- 4 Select the step that you want to run with option 1.
 - If you are configuring the interface for the first time, run every step.
 - If you do not need to run the wizard all the steps, press F4 to prompt the command, and then specify the number for the part of the interface you want to use.

- While you are running the wizard, use F12 to return to the previous panel. Use F3 to stop the wizard.

5 Press Enter to start the configuration process.

1: Read the Overview information (*OVERVIEW)

Setup JDE Interface	Overview
<u>Support for JDE IBM i Objects</u>	
Unique to JDE applications is a JDE repository (SVR), used to control compile time parameters for IBM i objects. There are special JDE-supplied programs used to create JDE objects. These programs perform processing prior to running the actual IBM i compile commands. Special type codes are provided (CBLJDE, CLPJDE, DSPJDE, LFJDE, PFJDE, PRTJDE, RPGJDE) for managing JDE objects. The create commands behind each of these type codes call the JDE programs that retrieve the necessary information from the JDE repository and then compile the programs, using standard IBM i compiler commands.	
The Set Up JDE Interface command verifies that the TURNOVER Lifecycle Manager environment is defined with the correct library list so that the JDE-supplied compile programs are available at promotion time. The command also verifies that the JDE type codes have been correctly defined and have been included in the JDE application(s).	
<u>Automatic SVR Update</u>	
There is an option to provide automatic update of SVR header and detail records throughout the change management cycle.	
F3=Exit	F12=Cancel
F21=System command	Enter=Continue
More...	

Read the Overview, and then press Enter. An analysis of the TURNOVER® Lifecycle Manager environment begins.

2: Check the TURNOVER® Lifecycle Manager environment (*ENV)

When the Check TURNOVER® Lifecycle Manager environment option is selected, choose either Traditional Environment Setup or Alternative Environment setup, and then press Enter.

Traditional environment setup

For TURNOVER® Lifecycle Manager to manage JD Edwards World objects, the environment library list must identify the World program library, the library that contains World control files, and the library that contains the World SVR files. The analysis attempts to locate these libraries (so it can add them to your environment library list) as follows:

Library	Searches for
JD Edwards World program library	QJDF data area
JD Edwards World control file library	F98301 file
JD Edwards World SVR file library	F9801 file

Depending on the choices made when the JD Edwards World application was installed, the control files might be in the common library, or the data library.

When it completes the analysis, the Set Up JDE Interface TURNOVER® Lifecycle Manager Environment panel appears.

Set Up JDE Interface
TURNOVER® Lifecycle Manager Environment

Environment . . . ENGLISH

Because the JDE interface references JDE-supplied programs and files, the TURNOVER® Lifecycle Manager environment must include the JDE program, common and SVR libraries as additional environment libraries.

JDE Program Library. . . .	JDFOBJ	Included in environment (usually JDFOBJ)
JDE Common Library	SLSCOM	Included in environment (CLTCOM)
JDE SVR Library.	SLSSEC	Included in environment (CLTSEC)

Environment configuration will support JDE release . . . 9.x

F3=Exit F12=Cancel F21=System command Enter=Continue

- 1 If TURNOVER® Lifecycle Manager did not find any of the libraries, it asks you to identify them. If TURNOVER® Lifecycle Manager finds the program, SVR, and/or control file libraries, it displays the information so you can verify it. (TURNOVER® Lifecycle Manager fills in the first valid library it finds.)
- 2 If the program or data library names are missing or incorrect, specify them here.
- 3 Press Enter. An analysis of the Programmer Worklist user-defined options begins.

Alternative environment setup

For TURNOVER® Lifecycle Manager to manage JD Edwards World objects, TURNOVER® Lifecycle Manager must be able to find the World program library, the library that contains World control files, and the library that contains the World SVR files. The alternative environment configuration panels are where you define to TURNOVER® Lifecycle Manager the libraries that are to be used to find the necessary JD Edwards objects. Each TURNOVER® Lifecycle Manager application that is used to manage JD Edwards objects can have its own alternative environment Configuration.

If an alternative environment configuration is not found for a specific TURNOVER® Lifecycle Manager application, the *DEFAULT alternate environment configuration will be used. A *DEFAULT alternative environment must be created, and the TSETUPJDE *ENV process is where the *DEFAULT alternative environment configuration is defined. Application-specific alternative environment configurations are defined in the TSETUPJDE *LIBLIST process.

The necessary JD Edwards files can be determined as follows:

- The JD Edwards World program library contains the data area QJDF.
- The JD Edwards World control file library contains the F98301 file.
- The JD Edwards World SVR file library contains the F9801 file.

At various points in TURNOVER® Lifecycle Manager processing, the current library list is exchanged with the libraries that in this list. For this process to work, you must also include QTEMP as the first library in this list, and QGPL as the last library in the list. These libraries should all be included in the *DEFAULT library list as shown in the screen example below:

```

Set Up JDE Interface
Alternative TURNOVER® Lifecycle Manager Environment

Application . . . *DEFAULT      Default Alternate Environment

Because the JDE interface references JDE-supplied programs and files, the
TURNOVER® Lifecycle Manager environment must include the JDE program library
(usually JDFOBJ), your common library (CLTCOM) and your SVR library (CLTSEC),
as well as QTEMP and QGPL.

Sequence  Library          Sequence  Library
  10      QTEMP             60      _____
  20      JDFOBJ            70      _____
  30      J73SLSCOM         80      _____
  40      J73SLSSEC         90      _____
  50      QGPL              100     _____

Alternative Environment configuration status . . . . . *ENABLED

Environment configuration will support JDE release . . . 9.x

F3=Exit   F12=Cancel   F21=System command   Enter=Continue
    
```

Once the *DEFAULT Alternative Environment Configuration contains all the correct libraries, press Enter to create the *DEFAULT records and then continue with the next step.

3: Check the programmer worklist user-defined options (*USEROPT)

You can use user-defined options from your Programmer Worklist to run JD Edwards World SVR options against World objects that are in the worklist.

The analysis checks to see if user-defined options for these JD Edwards World SVR functions already exist.

Function	Description
F02	Command line
F06	Repository services
F15	Where-used cross-reference
Option 8	Print source member
Option 9	Remove source member
Option 10	Design aid (SDA/RDA/FDA/KBG)
Option 20	Browse SDA/RDA

Function	Description
Option 25	Print Video, Report, or File Illustration

When the analysis completes, the Programmer Worklist User-Defined Options panel appears.

Set Up JDE Interface
Programmer Worklist User-Defined Options

The command TPWMJDEOPT can be executed, using Programmer Worklist user options, to run the following SVR maintenance options and function keys:

SVR Option	Description	PWM Option
F2	JDE command line	CL
F6	Repository Services	RS
F15	Where-used cross-reference	XR
Option 8	Print source member	PS
Option 9	Remove source member	O9
Option 10	Design Aid (SDA/RDA/FDA/KBG)	DA
Option 20	Browse SDA/RDA	BR
Option 25	Print Video, Report or File Illustration	PV

All of the above SVR options have been defined as Programmer Worklist user options. Press Enter to continue.

F3=Exit F12=Cancel F16=User options F21=System command Enter=Continue

- If the options are defined, they are listed with their associated option codes.
- If they do not exist, the option code column is blank, and you can create them.

1 Do one of the following:

- To create user-defined options for these functions, follow the prompts on the panel.

Press F16 to work with all the user-defined options on the worklist.

When you type option codes and press Enter, the command creates the worklist options for you, assigning them the codes you have specified.

- If you do not want to create options, leave the fields blank, and then press Enter to continue with the next panel.

You can create them later by rerunning this option, or by defining them manually (see ['2: Define programmer worklist user-defined options manually' on page 27](#)).

4: Check JD Edwards World application definitions (*APPL)

To manage JD Edwards World applications with TURNOVER® Lifecycle Manager, you must have one or more TURNOVER® Lifecycle Manager applications that are suitably configured for World applications.

The analysis checks your TURNOVER® Lifecycle Manager files for applications that have either "JDE" or "JD Edwards" in the application description, or that include type code DSPJDE.

The JDE Application Definitions panel appears.

Setup JDE Interface
JDE Application Definitions

No JDE application definitions have been detected. To create a JDE application definition based on a TURNOVER model application, select one of the following options:

1=Select

Create two level application. Level one promotes from development to test, and level two promotes from test to JDE-supplied production library.

1 Create three level application. Level one promotes from development to test, and level two promotes from test to user modifications library. Level three is the permanently locked JDE-supplied production library (recommended).

F3=Exit F12=Cancel F16=Applications F21=System command Enter=Continue

If JD Edwards World applications are found, they are listed for you to verify.

- 1 If the command cannot identify any JD Edwards World applications, do one of the following:
 - Select one of the options on the panel to create a two- or three-level JD Edwards World application, or
 - Press F16 to work with the application definitions on your system. You can adjust a set of existing application definition entries, or create a new set by copying one of the model applications (two-level model 20JD or three-level model 21JD).

When you return to the JDE Application Definitions panel, the analysis is performed again. If your application definition is recognized as a JD Edwards World application, it now appears on the panel.

- 2 Press Enter to continue. An analysis of your JD Edwards World application library lists begins.

5: Check application library lists (*LIBLIST)

Traditional environment setup

The library list for each JD Edwards World application must contain the names of the World program and data libraries, as they appear in the environment library list. (Remember, these libraries contain objects that are required for successfully compiling JD Edwards World source code.) The library list analysis checks to be sure the JD Edwards World applications identified by the preceding step have a library list, and that the list contains the appropriate World program and data library names.

The Application Library Lists panel appears.

```

Set Up JDE Interface
Application Library Lists

The following JDE libraries must be included in application library lists:

Program library . . . JDFOBJ
Common library. . . . SLSCOM
SVR library . . . . . SLSSEC
One or more of the following application library lists does not include the
above JDE libraries. Use option 1 next to each application to which you want
to add the required libraries. Use F16 to work with library lists.

1=Select

  Appl  Rel Ver Lev Description                               Library List
  --  --  --  --  --  --
  _ 20JD          1 Sample JD Edwards Appl  DEV --> TEST    20JD010101
  _ 20JD          2 Sample JD Edwards Appl  TEST --> PROD   20JD010102
  _ 21JD          1 Sample JD Edwards Appl  DEV --> TEST    21JD010101
  _ 21JD          2 Sample JD Edwards Appl  TEST --> MODS   21JD010102
  _ 21JD          3 Sample JD Edwards Appl  PERM LOCKED    21JD010103 +

F3=Exit  F12=Cancel  F16=Library lists  F21=System command  Enter=Continue
  
```

- 1 If a recognized JD Edwards World application uses a library list that does not contain the program and data libraries shown, do one of the following:
 - Press Enter to have the command add them to the library list.
 - Use F16 to work with TURNOVER® Lifecycle Manager library list objects, where you can add them yourself.
- 2 To change a TURNOVER® Lifecycle Manager library list object, you must be authorized to *all* the applications that use it.

To see a list of all the applications that use a library list, press F16, select a list with option 7. Then use option 3 on the main menu to change application authority if necessary.
- 3 Include `QTEMP` at the beginning of the library list, and `QGPL` at the end of the library list. TURNOVER® Lifecycle Manager does not do this automatically, so you must do it manually.
- 4 When you are satisfied that the JD Edwards World application library lists for this environment are in order, press Enter.

An analysis of the type codes for your JD Edwards World applications begins.

Alternative environment setup

The library list for each JD Edwards World application must contain the names of the World program and data libraries, as they appear in the corresponding Alternative Environment Configuration library list. (These libraries contain objects that are required for successfully compiling JD Edwards World source code.) The library list analysis checks that:

- the JD Edwards World applications identified by the preceding step have a library list
- the list contains the appropriate World program and data library names.

The Application Library Lists appears, with the results of the analysis.

- If an JD Edwards World application has an alternative environment configuration, the library list will be checked against that configuration.
- If an application does not have an alternative environment configuration defined, **TURNOVER® Lifecycle Manager** runs further checks on the library list against the *DEFAULT alternative environment configuration.

- 1 If an application uses a library list that does not contain the program and data libraries shown, do one of the following:
 - Press Enter to add them to the library list.
 - Press F16 to work with **TURNOVER® Lifecycle Manager** library list objects, where you can add them yourself.
- 2 To change a **TURNOVER® Lifecycle Manager** library list object, you must be authorized to *all* the applications that use it. To see a list of all the applications that use a library list, press F16, and then select a list with option 7. Then use option 3 on the main menu to change application authority.
- 3 Include **QTEMP** at the beginning of the library list, and **QGPL** at the end of the library list. **TURNOVER® Lifecycle Manager** does not do this automatically, so you must do it manually.

To create an application-specific alternative environment configuration

- 1 Type 6 next to the application, and then press Enter.

The Alternative Environment panel appears.

```

Set Up JDE Interface
Alternative TURNOVER® Environment

Application . . . 20JD/0/0      Sample JD Edwards Application 1

Because the JDE interface references JDE-supplied programs and files, the
TURNOVER environment must include the JDE program library (usually JDFOBJ),
your common library (CLTCOM) and your SVR library (CLTSEC), as well as QTEMP
and QGPL.

      Sequence  Library           Sequence  Library
      10      QTEMP             60      _____
      20      JDFOBJ            70      _____
      30      CLTCOM            80      _____
      40      CLTSEC            90      _____
      50      QGPL             100     _____

Alternative Environment configuration status . . . . . *DISABLED

Environment has not been verified to support the JDE interface.
F3=Exit  F12=Cancel  F21=System command  Enter=Continue
Alternative Environment record not found.  Press ENTER to create.
  
```

This panel shows the JD Edwards libraries that should be included, based on the application library list.

- 2 You can accept the suggested libraries and press Enter, or change the libraries to other values and press Enter to create the Alternative Environment Configuration (see Alternative Environment Setup under *ENV for more information on this panel).
- 3 Press Enter to continue.

An analysis of the type codes for your JD Edwards World application begins.

6: Check type code definitions (*TYPECODES)

An application to handle JD Edwards World objects must have the appropriate type codes for all the World objects it will manage. This analysis checks to see whether the application contains any of the JD Edwards World type codes, and examines the create command specified for each type code to be sure it is correct.

It is easiest to include all the JD Edwards World type codes in every World application, and then use only the ones you need.

```

Set UP JDE Interface
Type Code Definitions

The following type codes are used to manage JDE objects:

IBM i object types . . . . . CBLJDE, CLPJDE, DSPJDE, LFJDE, PFJDE, PRTJDE,
                              RPGJDE
Data-driven object types . . . JDEDD, JDEDW, JDEWV, JDEMNU, JDEPO, JDEUDC,
                              JDEVO, JDEWV, JDEFSV

One or more of the above type codes is either missing or incorrectly defined
for one or more of the following applications. Use option 1 to add the
missing type codes and/or correct the type code definitions for the selected
applications, or press F16 to work with applications.

  Appl  Rel Ver Lev Description
  --  --  --  --  --
  - DUG2   2      1  *PGMR to production
  - 20JD      1  Sample JD Edwards Appl  DEV --> TEST
  - 20JD      2  Sample JD Edwards Appl  TEST --> PROD
  - 21JD      1  Sample JD Edwards Appl  DEV --> TEST
  - 21JD      2  Sample JD Edwards Appl  TEST --> MODS
                                     +

F3=Exit  F12=Cancel  F16=Applications  F21=System command  Enter=Continue
Application 20JD/00/00/01 missing type codes: CBLJDE JDEWV JDEFSV
                                     +
  
```

If any of the JD Edwards World applications are missing World type codes, or if they are configured incorrectly, the applications are highlighted, and a message appears a message at the bottom of the panel to indicate the errors.

- 1 Review the errors, type 1 next to the applications to which you want to add the type codes, and then press Enter. TURNOVER® Lifecycle Manager adds or corrects the type codes.
- 2 After TURNOVER® Lifecycle Manager completes its processing, some of the applications might still have type code errors. Review the errors at the bottom of the panel to determine what needs adjustment. Press F16 to edit the applications manually. A message does not necessarily indicate an error.
- 3 Press Enter to continue with the next step.

7: Check for JD Edwards World data-driven definition files (*JDEFILES)

To promote JD Edwards World data-driven objects up through the levels of your World application, copies of the definition files must exist in each level's target libraries. The analysis checks each JD Edwards World application that uses World data-driven type codes, to make sure its target libraries contain the appropriate definition files.

After the analysis completes, the JDE Data-Driven Files panel appears.

```

Set UP JDE Interface
JDE Data-Driven Files

The following type codes are use manage data-driven objects:

Data-driven object types . . . JDEDD, JDEDW, JDEDWV, JDEMNU, JDEPO, JDEUDC,
JDEVO

One or more defining files for the above type codes do not exist for one or
more of the following applications. Press Enter to create the missing files
for the selected applications.

1=Select

  Appl  Rel Ver Lev Description
  --  --  --  --  --
  -- 20JD      1 Sample JD Edwards Appl DEV --> TEST
  -- 20JD      2 Sample JD Edwards Appl TEST --> PROD
  -- 21JD      1 Sample JD Edwards Appl DEV --> TEST
  -- 21JD      2 Sample JD Edwards Appl TEST --> MODS

F3=Exit  F12=Cancel  F16=Applications  F21=System command  Enter=Continue
+
    
```

If any of the JD Edwards World definition files is missing from any of your World application definitions, a message appears for each definition file that is missing. (You can see all the messages by moving your cursor to the message area and scrolling up or down.) Do one of the following:

- To copy the missing files into the target library, select an application with 1, and then press Enter.
 - To manually adjust an application's target library, press F16.
- 1 When you select an application with 1 to indicate you want TURNOVER® Lifecycle Manager to create the files, TURNOVER® Lifecycle Manager displays a list of the libraries from which it will duplicate the files, and the files it will create and the libraries in which it will create them.
 - 2 Use JDE options to create additional JDE environments. If you are missing files from a specified library, it could indicate that your JDE software is not setup correctly. You should use this option to create files only in rare cases.
 - 3 When you are satisfied that the correct files are in the target libraries, press Enter.

8: Check SVR automatic update (*SVR)

TURNOVER® Lifecycle Manager can update both header and detail records in the SVR whenever JD Edwards World objects are checked out, promoted, or distributed. The analysis checks these conditions:

- The global type code JDESV must have an object type of *JDESVAUTO.
- The JDESV type code must be in the type code list for any JD Edwards World application that should use automatic SVR updating.
- The JDESV application type code at each application level must specify the library that contains the SVR header file (F9801) and detail file (F9802) as the target object library.

Check phases

These checks occur in two phases:

- 1 The first phase checks for an object type of *JDESVAUTO in the JDESV global type code definition.

If the type code definition needs to be changed, the SVR Automatic Update panel appears.

Set UP JDE Interface
SVR Automatic Update

For the SVR to be automatically updated, the global type code JDESV must be defined with an object type of *JDESVAUTO. Press Enter to update the global type code, or press F12 to cancel.

F3=Exit F12=Cancel F16=Applications F21=System command Enter=Continue

- 2 To update the global type code, press Enter.

The second phase of the analysis starts.

The analysis assumes that the JDESV global type code already exists on your system. This should be the case, because type codes are created when you install TURNOVER® Lifecycle Manager, and refreshed when you upgrade or apply a cumulative change tape. However, if the type code is deleted, you must create it before the analysis can continue: see ['Missing JD Edwards World type codes' on page 33](#).

Each TURNOVER® Lifecycle Manager application should update only one SVR. Do *not* have multiple SVRs defined to a single application. If you have F9801 or F9802 SVR files in your Development or QA libraries, you may not have your JD Edwards software setup correctly: contact UNICOM Systems, Inc. technical support for further guidance.

The second phase of the analysis handles requirements 2 and 3 (the application type codes and the target library for SVR records).

If any of your JD Edwards World applications do not meet the application type code and target library requirements, TURNOVER® Lifecycle Manager lists them.

```

Set UP JDE Interface
SVR Automatic Update

All applications should update the SVR in the same library. Use option 1
to have the selected applications update the SVR in the library shown below,
or press F12 to cancel.

SVR library . . . SLSDTA

1=Select

  Appl  Rel Ver Lev Description                               SVR Library
  1  20JD          1 Sample JD Edwards Appl  DEV --> TEST
  1  20JD          2 Sample JD Edwards Appl  TEST --> PROD
  1  21JD          1 Sample JD Edwards Appl  DEV --> TEST
  1  21JD          2 Sample JD Edwards Appl  TEST --> MODS
  1  21JD          3 Sample JD Edwards Appl  PERM LOCKED          +

F3=Exit  F12=Cancel  F16=Applications  F21=System command  Enter=Continue
  
```

3 To edit the type codes, do one of the following:

- To update the application JDESV type code with the suggested SVR library, select the application levels with 1, and then press Enter.
- To edit the applications' JDESV type code manually, press F16. For more information, see [7: Configure automatic SVR update manually' on page 36](#).

4 When you finish, a message appears indicating that the applications have been configured for *automatic* SVR updating. Press Enter.

9: Check remote automatic distribution jobs (*TURCRARE)

Make this change only if you are using the TURCRARE autostart job to receive distributions on your remote systems. If you receive distributions manually, this step does not apply.

This panel is for information only.

```

Set Up JDE Interface
Check Jobd for TURCRARE on Remote Systems

If you are distributing using SNADS to remote systems, and you are using
autostart job TURCRARE on the remote systems to automatically receive
distributions, the JDE program, common and SVR libraries must be included
in the initial library list of Jobd TURCRARE in the TURNOVER product library
on the remote systems.

The Set Up JDE Interface command does not verify the TURCRARE Jobd on the
remote systems. You must do this manually.

To verify the TURCRARE Jobd on a remote system, sign on to that system and
use the DSPJOB command to display the TURCRARE Jobd. The Initial Library
List must include the JDE product, common and SVR libraries. If these
libraries are not included in the initial library list, use the CHGJOB command
to update the initial library list for Jobd TURCRARE.

Press Enter to continue.

F3=Exit  F12=Cancel  F21=System command  Enter=Continue
  
```

For more information, see *'Changing the remote TURCRARE job descriptions' on page 37.*

10: Enable alternate environments (*USEALTENV)

This step applies only to the alternative environment configuration, and there are no panels associated with this step.

When this option is taken, TURNOVER® Lifecycle Manager check that:

- you have defined a *DEFAULT alternative environment
- all alternative environments contain a JD Edwards program library, control library and SVR library
- the SVR update information in the JDE applications has been defined correctly.

Note This option validates ALL defined JDE applications, whether you have authority to them or not. Alternate environments are enabled only if ALL JDE applications are properly set up. If you do not have authority to all JDE applications, and enablement fails, then someone authorized to those applications must make the necessary adjustments before alternate environments can be enabled.

Once all checks have been passed, the alternative environment configurations are *ENABLED for use.

Setting up the interface manually

This section describes how to manually configure TURNOVER® Lifecycle Manager to manage JD Edwards World objects.

If you are using the alternative environment configurations, some of these steps must be performed through the setup wizard (TSETUPJDE), or you must use the TSETUPJDE wizard used after you have made manual adjustments.

1: Configure the TURNOVER® Lifecycle Manager environment manually

TURNOVER® Lifecycle Manager uses JD Edwards World object create commands, definition files, and SVR records at various stages of the change process. For TURNOVER® Lifecycle Manager to see and use these items, the JD Edwards World libraries which they are must be identified in TURNOVER® Lifecycle Manager's environment library list.

Note These instructions apply to only a traditional environment setup. Alternative environment configuration is available only by using the TSETUPJDE wizard.

- 1 Sign on as QSECOFR.
- 2 On a TURNOVER® Lifecycle Manager command line, type WRKTOENV, and then press Enter.

- 3 Select the environment where you want to manage JD Edwards World objects, and then press Enter.

The Change an Environment panel appears.

MM/DD/YYYY 14:46:24	Change an Environment	Your Company, Inc. YOURSYS
Type required Environment information and press Enter.		
Environment	ENGLISH	
Description	English Production Version	
Product library	SOFTTURN	
Data library	SOFTTURN	Name
Language library	SOFTTURN	Name
Message file	TNLENG	Name
Coded character set ID		
Additional environment libraries . .		
	JDFOBJ	
	CLTDTA	
	CLTCOM	
	CLTSEC	
F3=Exit F12=Cancel		

- 4 Under **Additional environment libraries**, specify the names of the JD Edwards World program object, data, common, and security libraries, and then press Enter.

What libraries you specify here depends on the options you chose when you installed the JD Edwards World application on your IBM i system. TURNOVER® Lifecycle Manager must be able to find the SVR, the JD Edwards World program objects, and the control files for your World data-oriented objects (such as Dream Writers, Vocabulary Overrides, and so on). You can specify other JD Edwards World data file libraries in your TURNOVER® Lifecycle Manager World application definition.

2: Define programmer worklist user-defined options manually

All of the JD Edwards World IBM i and data driven object types can be managed on the programmer worklist. Checkout, editing, object creation, cross-referencing, and source compare are all supported when you select your JD Edwards World objects with various standard TURNOVER® Lifecycle Manager processing options. For more information about the programmer worklist, see *'Managing JD Edwards World objects from a programmer worklist' on page 68*.

In addition to the standard TURNOVER® Lifecycle Manager options, you can create user-defined options that perform JD Edwards World SVR functions against World objects on your worklist.

Suggested user-defined worklist options for JD Edwards World development

The PWM codes are suggestions. You can use any codes you choose.

SVR function	Description	PWM option	Command to use for worklist option
F02	Command line	CL	TPWMJDEOPT OBJ("&OB") LIB("&OL") ATTR("&OA") SRCF("&PF") SRCFLIB("&PL") JDEOPT(*FKY02)
F06	Repository services	RS	TPWMJDEOPT OBJ("&OB") LIB("&OL") ATTR("&OA") SRCF("&PF") SRCFLIB("&PL") JDEOPT(*FKY06)
F15	Where-used cross-reference	XR	TPWMJDEOPT OBJ("&OB") LIB("&OL") ATTR("&OA") SRCF("&PF") SRCFLIB("&PL") JDEOPT(*FKY15)
Option 8	Print source member	PS	TPWMJDEOPT OBJ("&OB") LIB("&OL") ATTR("&OA") SRCF("&PF") SRCFLIB("&PL") JDEOPT(*OPT08)
Option 9	Remove source member	O9	TPWMJDEOPT OBJ("&OB") LIB("&OL") ATTR("&OA") SRCF("&PF") SRCFLIB("&PL") JDEOPT(*OPT09)
Option 10	Design aid (SDA/RDA/FDA/KBG)	DA	TPWMJDEOPT OBJ("&OB") LIB("&OL") ATTR("&OA") SRCF("&PF") SRCFLIB("&PL") JDEOPT(*OPT10)
Option 20	Browse SDA/RDA	BR	TPWMJDEOPT OBJ("&OB") LIB("&OL") ATTR("&OA") SRCF("&PF") SRCFLIB("&PL") JDEOPT(*OPT20)
Option 25	Print Video, Report, or File Illustration	PV	TPWMJDEOPT OBJ("&OB") LIB("&OL") ATTR("&OA") SRCF("&PF") SRCFLIB("&PL") JDEOPT(*OPT25)

Given this information, create the user-defined worklist options as follows:

- Do one of the following:
 - On the TURNOVER® Lifecycle Manager main menu, choose option 12.
 - On a TURNOVER® Lifecycle Manager command line, type STRPWM.
- Select any worklist, and then press Enter.
- Press F16 to work with user-defined options for the Programmer Worklist.

The Maintain User Defined Options for Programmer Worklist Manager panel appears.

```

MM/DD/YYYY          Maintain User Defined Options          Your Company, Inc.
9:53:44             For: Programmer Worklist Manager

Protected options already exist as TURNOVER® options.          File: TUSROPTF
                                                                Library: *LIBL

2=Change option    4=Delete option

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

F3=Exit    F6=Add option    F11=Alt Vw    F12=Cancel
    
```

- 4 Scroll through the list to see the option codes that exist.

You might need to change the ones that are mentioned in *'Suggested user-defined worklist options for JD Edwards World development' on page 27*.

- 5 To create a new user-defined option, press F6.

The Add User Defined Options panel appears.

```

MM/DD/YYYY          Add User Defined Options          Your Company, Inc.
11:28:39            For: Programmer Worklist          YOURSYS

Type required information and press enter.

Option . . . . . J1
Description . . . . JDE Cmd Line

Create command . . . TPWMJDEOPT OBJ("&OB") LIB("&OL") ATTR("&OA") SRCF("&PF")
SRCFLIB("&PL") JDEOPT(*FKY02)

"&A" =Application          "&PF"=Production Source File
-11 =Release                "&PL"=Production Source Lib
-22 =Version                "&PM"=Production Source Mbr
-33 =Level                  "&OB"=Object Name
"&F#"=Form Number           "&OL"=Object Library
"&PG"=Programmer            "&OA"=Object Attr (PF, RPG)
"&RF"=Project Reference

F3=Exit    F4=Prompt command    F12=Cancel
    
```

This example screen shows the definition for SVR function key F2 as an example.

For each option

For each option you define, do the following:

- 1 In **Option**, type an option code.
- 2 In **Description**, type a description of the option's function.

- 3 In **Create command**, type the command that corresponds with the option you are defining: see *'Suggested user-defined worklist options for JD Edwards World development' on page 27*.
- 4 Press Enter.
- 5 Repeat these steps to define the remaining options, using the information in *'Suggested user-defined worklist options for JD Edwards World development' on page 27*.
- 6 When you finish, press F3.

The list of options now includes the JD Edwards World options you have defined. Press F3 until you get back to the main menu.

3: Create JD Edwards World application definitions manually

You are recommended to use a three-level permanently locked application for managing JD Edwards World applications. TURNOVER® Lifecycle Manager includes two models: a three-level application (21JD) and a two-level application (20JD). Review the models to make sure you understand them.

The two-level application might be useful on your remote systems, if you test distributed changes there before promoting them to production.

Note The quickest way to create a complete JD Edwards World application definition is by creating a new one based on one of the World models, using the `CRTAPPDEFN` command.

If you want more levels in your JD Edwards World application, you can still follow these instructions for the three-level application. Copy and modify one level as needed to create your additional levels.

Each level of your JD Edwards World application must have the following fundamental elements:

- appropriate **From** and **Target** source, object, data, and JD Edwards World control file library names (Define Application Libraries panel)
- an appropriate library list
- JD Edwards World object type codes for all World objects the application must handle (Type Codes panel).

The `CRTAPPDEFN` method encompasses these requirements by taking you on a “guided tour” of the application definition process.

4: Create a JD Edwards World application definition

- 1 On the TURNOVER® Lifecycle Manager main menu, select option 1 to work with application definitions, and then press F6.

The Create Application Definition panel appears.

- 2 Type an application code:

- To create a three-level application, specify 21JD.
- To create a two-level application, specify 20JD.

3 Accept the remaining defaults, and then press Enter.

Thee Application Libraries panel appears.

MM/DD/YYYY		Define Application Libraries		Your Company, Inc.	
14:50:18				YOURSYS	
Application: MJDE					
	Data Obj.	Other Obj.	Source	JDE	
Lev	Libraries	Libraries	Libraries	Control File	
	*PGMR	*PGMR	*PGMR	MJDEVCOM	
1	MJTESTDTA	MJTESTOBJ	MJTESTSRC	MJTESTCOM	
2	MJMODSDTA	MJMODSOBJ	MJMODSSRC	MJMODSCOM	
3	JDFDATA	JDFOBJ	JDFSRC	JDFDATA	

F3=Exit F12=Cancel F21=System entry

4 Type the names of the appropriate libraries for each application level.

For applications that will use the JD Edwards World data-driven type codes, you must type the library name or use *PGMR; you cannot use *PROJECT.

5 Press Enter.

The Define Library List Objects panel appears.

5: Create library list objects for your JD Edwards World application

Setting up library lists for your application occurs automatically. Review them to make sure they are correct.

MM/DD/YYYY	Define Library List Objects		Your Company
13:59:49			YOURSYS
Application: <u>MJDE</u> Rel: Ver:			
2=Change library list			
Lev	Library List	Description	
— 1	MJDE000001	Application MJDE Release 00 Version 00 Level 01	
— 2	MJDE000002	Application MJDE Release 00 Version 00 Level 02	
F3=Exit F12=Cancel F21=Command line			

- 1 On the Define Library List Objects panel, select the library list objects you want to review with option 2.

Add `QTEMP` to the beginning of each library list object, and to add `QGPL` to the end.
 TURNOVER® Lifecycle Manager requires these to properly process JDE objects.

- 2 When you are satisfied with the contents of the library lists, press Enter.
- 3 When you have reviewed the library list objects for all levels, press Enter to accept them.

The Define the Application panel for the first application level appears, with options pre-selected based on the model you used.

- 4 When you press Enter, the panel or panels associated with each pre-selected option appear in sequence so you can customize the application level according to your development rules and requirements.

After each pre-selected panel is completed, the Define the Application panel appears again.

You can toggle to the next level of the application using F18.

- 5 Finally, the Confirm Initial Build and Create Missing Items panels appear, to build the application definitions and create any missing files and libraries that they require.

6: Include the JD Edwards World type codes in your application definition

This section applies only if you did not use `CRTAPPDFN` to build your JD Edwards World application. If you used `CRTAPPDFN`, go to [7: Configure automatic SVR update manually' on page 36.](#)

You must ensure that each level of your JD Edwards World application includes the World type codes in its list of valid object type codes.

A TURNOVER® Lifecycle Manager JD Edwards World application definition needs special type codes for each kind of World object included in your World application. There are two sets of special global type codes: one for JD Edwards World IBM i objects, and one for World data-driven objects.

You should not have to create global type codes for JD Edwards World objects, because they are already in the global type code file.

To verify the type codes

- 1 On the main menu, select option 1. On the Work with Application Definitions panel, select your application with option 2 (Change), and then press Enter.
- 2 Select Type codes with option 1, and then press Enter.
- 3 Make sure all the JD Edwards World type codes shown in the tables appear in the list.

If any are missing, press F4 to select more codes from the global type code file.

If you cannot find one or more codes in the global type code list, see ['Missing JD Edwards World type codes' on page 33](#), and then create the codes you need. Return to the application definition, and then include the recreated codes in the type code list at all application levels, as instructed here.

- 4 When you finish selecting codes from the global list, press Enter, and then press F3 to get back to the Define the Applications panel.
- 5 Press F18 to go to the next level of the application so you can check its type code list.
- 6 Repeat steps 2 through 5 until you have checked the type code list for every application level.
- 7 Press F3 until you reach the main menu, and then continue with ['7: Configure automatic SVR update manually' on page 367](#).

Missing JD Edwards World type codes

When you update TURNOVER® Lifecycle Manager, the type codes are refreshed to keep them up-to-date.

If you need to recreate or correct a JD Edwards World type code use the information below.

Type code specifications for JD Edwards World IBM i objects

Type code	Create command	Object type	Seq.	Attribute	Not e
CBPJDE	SOFTTURN/JDECBL MEMBER("&OB") FILE("&SF") OBJLIB("&LI") SRCLIB("&SL")	*PGM	500	CBL	
CLPJDE	SOFTTURN/JDECLP MEMBER("&OB") FILE("&SF") OBJLIB("&LI") SRCLIB("&SL")	*PGM	500	CLP	

Type code	Create command	Object type	Seq.	Attribute	Not e
DSPJDE	SOFTTURNE/JDEDSPF MEMBER("&OB") FILE("&SF") OBJLIB("&LI") SRCLIB("&SL")	*FILE	450	DSPF	
LFJDE	SOFTTURNE/JDELF MEMBER("&OB") FILE("&SF") OBJLIB("&LI") SRCLIB("&SL")	*FILE	425	LF	
PFJDE	SOFTTURNE/JDEPF MEMBER("&OB") FILE("&SF") OBJLIB("&LI") SRCLIB("&SL")	*FILE	410	PF	
PRTJDE	SOFTTURNE/JDEPRTF MEMBER("&OB") FILE("&SF") OBJLIB("&LI") SRCLIB("&SL") COVER(1)	*FILE	440	PRTF	
RPGJDE	SOFTTURNE/JDERPG MEMBER("&OB") FILE("&SF") OBJLIB("&LI") SRCLIB("&SL")	*PGM	500	RPG	1

Type code specifications for JD Edwards World data-driven objects

Type code	Create command	Object type	Seq.	Attribute	Not e
JDEDD	SOFTTURNE/CPYJDEOBJ OBJ("&SM") OBJTYPE(*JDEDD) FROMLIB("&FL") TOLIB("&SL") CPYOPT(*REPLACE)	*JDEDD	100	JDEDD	2
JDDSMP	SOFTTURNE/CPYJDEOBJ OBJ("&SM") OBJTYPE(*JDDl1ssss) FROMLIB("&FL") TOLIB("&SL") CPYOPT(*REPLACE)	*JDDl1ssss	100	JDDSMP	2
JDEDW	SOFTTURNE/CPYJDEOBJ OBJ("&SM") OBJTYPE(*JDEDW) FROMLIB("&FL") TOLIB("&SL") CPYOPT(*REPLACE)	*JDEDW	100	JDEDW	3
JDEDWV	SOFTTURNE/CPYJDEOBJ OBJ("&SM") OBJTYPE(*JDEDWV) FROMLIB("&FL") TOLIB("&SL") CPYOPT(*REPLACE)	*JDEDWV	100	JDEDWV	3
JDWSMP	SOFTTURNE/CPYJDEOBJ OBJ("&SM") OBJTYPE(*JDWpppppp) FROMLIB("&FL") TOLIB("&SL") CPYOPT(*REPLACE)	*JDWpppppp	100	JDWSMP	
JDEMNU	SOFTTURNE/CPYJDEOBJ OBJ("&SM") OBJTYPE(*JDEMNU) FROMLIB("&FL") TOLIB("&SL") CPYOPT(*REPLACE)	*JDEMNU	100	JDEMNU	4
JDEPO	SOFTTURNE/CPYJDEOBJ OBJ("&SM") OBJTYPE(*JDEPO) FROMLIB("&FL") TOLIB("&SL") CPYOPT(*REPLACE)	*JDEPO	100	JDEPO	7
JDESV	SOFTTURNE/CPYJDEOBJ OBJ("&SM") OBJTYPE(*JDESV) FROMLIB("&FL") TOLIB("&SL") CPYOPT(*REPLACE)	*JDESVAUT O	100	JDESV	8

Type code	Create command	Object type	Seq.	Attribute	Not e
JDEUDC	SOFTTURN/CPYJDEOBJ OBJ ('"&SM"') OBJTYPE(*JDEUDC) FROMLIB("&FL") TOLIB("&SL") CPYOPT(*REPLACE)	*JDEUDC	100	JDEUDC	5
JUDSMP	SOFTTURN/CPYJDEOBJ OBJ ('"&SM"') OBJTYPE(*JUD11) FROMLIB("&FL") TOLIB("&SL") CPYOPT(*REPLACE)	*JUD11	100	JUDSMP	
JDEVO	SOFTTURN/CPYJDEOBJ OBJ ("&SM") OBJTYPE(*JDEVO) FROMLIB("&FL") TOLIB("&SL") CPYOPT(*REPLACE)	*JDEVO	100	JDEVO	6
JVOSMP	SOFTTURN/CPYJDEOBJ OBJ ("&SM") OBJTYPE(*JVOLLSSSS) FROMLIB("&FL") TOLIB("&SL") CPYOPT(*REPLACE)	*JVOLLSSSS	100	JVOSMP	6
JDEWWV	SOFTTURN/CPYJDEOBJ OBJ ("&SM") OBJTYPE(*JDEWWV) FROMLIB("&FL") TOLIB("&SL") CPYOPT(*REPLACE)	*JDEWWV	100	JDEWWV	3
JDEFSV	SOFTTURN/CPYJDEOBJ OBJ ("&SM") OBJTYPE(*JDEFSV) FROMLIB("&FL") TOLIB("&SL") CPYOPT(*REPLACE)	*JDEFSV	100	JDEFSV	3

Notes

These notes apply to both of the tables above.

- You can use the JDERPG command to create OPM or ILE programs.

For more information, see *'Create JD Edwards RPG Program command: JDERPG' on page 43.*

- The JDD Data Dictionary type supports languages and systems.

For more information, see *'Language, system, and version support (for JDEDD, JDEDW, JDEWWV, JDEFSV, and JDEUDC)' on page 62.*

- For more information about Dream Writers, World Writers, and FASTRs, see:

'Language, system, and version support (for JDEDD, JDEDW, JDEWWV, JDEFSV, and JDEUDC)' on page 62

'Managing Dream Writer, World Writer, and FASTR versions' on page 52.

- The JDEMNU Menu type supports multiple languages.

See *'Menu language preferences' on page 75.*

- Quotes are used differently for the JDEUDC type code's OBJ parameter. Also, the JDEUDC User Defined type supports languages.

For more information about JDEUDC types, see *'Language, system, and version support (for JDEDD, JDEDW, JDEWWV, JDEFSV, and JDEUDC)' on page 62.*

- The JDEVO Vocabulary Override type supports multiple languages and systems.

For more information about JDEVO types, see *'More about JD Edwards World vocabulary overrides (JDEVO)' on page 56*.

- For more information about processing options, see *'Checking out versions' on page 56*.
- TURNOVER® Lifecycle Manager automatically updates your SVR and uses the JDESV type code for information about where your SVR resides.
- The promotion method for the type codes for data-driven objects must be CSCO.

7: Configure automatic SVR update manually

TURNOVER® Lifecycle Manager can update your application's Software Version Repository (SVR) automatically any time a JD Edwards World object is checked out or promoted. To make this happen, *each level* of your JD Edwards World application must meet the following conditions:

- The application type code list must include type code JDESV.

See *'6: Include the JD Edwards World type codes in your application definition' on page 32*.

- The JDESV global type code must have an object type of *JDESVAUTO.

The type code is shipped with the **Object type** set at this value. To verify it, see *'Checking the JDESV object type' on page 36*.

- The JDESV application type code for all levels of your JD Edwards World application must specify the same target object library (usually your security library, CLTSEC) where the SVR files (F9801 and F9802) are.

To verify it, see *'Checking the target object library for JDESV' on page 36*.

Checking the JDESV object type

- 1 On the TURNOVER® Lifecycle Manager main menu, select option 8, and then select option 4.
- 2 In the **Position to** field, type JDESV, and then press Enter.

You should see the JDESV type code in the list, and its object type should be *JDESVAUTO. If the type code has been changed or deleted, you can re-define it: see *'Type code specifications for JD Edwards World data-driven objects' on page 34*.

Checking the target object library for JDESV

- 1 On the TURNOVER® Lifecycle Manager main menu, select option 1 to work with applications, and then press Enter.
- 2 Select your World application with option 2 and press Enter.
- 3 Select **Type codes** with option 1, and then press Enter.
- 4 Select the JDESV type code with option 2, and then press Enter.

Make sure the name of the **Target object library** is correct.

- 5 Do this for all levels of your World application to make sure the **Target object library** is *the same* for all levels.
- 6 If necessary, do this for all your World applications.

Changing the remote TURCRARE job descriptions

If you distribute changed JD Edwards World objects to other systems and you are using the TURCRARE autostart job to process the distributed changes, you must make sure that the TURCRARE job description on those systems has the World program and data libraries in its initial library list.

The TURCRARE job description libraries must include the same JD Edwards World libraries as you put in your TURNOVER® Lifecycle Manager environment library list.

To check the TURCRARE job description on each remote system

- 1 Sign on to the remote system as QSECOFR.
- 2 Type `DSPJOB prodlib/TURCRARE`, and then press Enter.

Replace *prodlib* with the name of your system's TURNOVER® Lifecycle Manager product library.
- 3 Press F10 to see additional parameters, and then press Page Down to go to the **Initial library list** field.
- 4 Do one of the following:
 - If the JD Edwards World program and data libraries are in this list, no change is required. Press F3 to exit.
 - If they are not in the list, press Enter to end the `DSPJOB` command.
- 5 On a command line, type `CHGJOB prodlib/TURCRARE`, and then press F4.

Replace *prodlib* with the name of your system's TURNOVER® Lifecycle Manager product library.
- 6 Press F10, and then press Page Down to go to the **Initial library list** field.
- 7 Use the **+ for more values** field.

The Specify More Values for Parameter INLLIBL panel appears:

Specify More Values for Parameter INLLIBL

Type choices, press Enter.

Initial library list	SOFTTURN	Name, *SAME, *SYSVAL, *NONE
	JDFOBJ	
	CLTDTA	
	CLTSEC	
	CLTCOM	

More...

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

- 8 Type the names of your JD Edwards World object library, client data library, client common library, and client security library, and then press Enter.

Reviewing the JD Edwards World application type codes

TURNOVER® Lifecycle Manager supports JD Edwards World object create programs and data-driven object movement programs. You can use the JD Edwards World commands and override their parameters occasionally, when exceptions occur.

TURNOVER® Lifecycle Manager supports these commands through type codes. There are type codes that control creation of standard IBM i JD Edwards World objects, and type codes that control the movement of the data-driven World objects. One aspect of setting up TURNOVER® Lifecycle Manager to manage JD Edwards World objects is adding the appropriate TURNOVER® Lifecycle Manager World type codes to each level of your World application definitions.

All of the JD Edwards World model application definitions include all of the World type codes. Review all levels of the model applications before you begin.

- 1 Do one of the following:
 - On the TURNOVER® Lifecycle Manager main menu, select option 1 (Work with Application definitions).
 - On a TURNOVER® Lifecycle Manager command line, type `WRKAPPDFN`, and then press Enter.
- 2 Select the World application model with option 2, and then press Enter.

```

                                Define the Application
                                Your Company, Inc.
                                SYSTEM: YOURSYS
                                Action: CHANGE

Application . . . 21JD JDE Application- Test Level 1
Release . . . . . Lock . . . . . No
Version . . . . . XRef . . . . . *TURNOVER
Level . . . . . 1 Fastturn . . . No
Type options, press Enter. Press F21 to select all.
  1=Select
    _ Description
    _ Application defaults and rules
    _ Project defaults
    _ Library defaults
    _ Form defaults
    _ Line defaults
    1 Type codes
    _ Authorized users
    _ Distribution defaults
    _ Approval list
    _ Form completion messages
    _ Application relations
    _ Version and stamping defaults

F3=Exit  F12=Cancel                                F18=Next level  F21=Select all
  
```

3 Select **Type codes**.

The Type Codes panel appears.

```

MM/DD/YYYY                                Type Codes                                Your Company, Inc.
13:58:10      Application: 20JD Rel:      Ver:      Lev: 1      SYSTEM: YOURSYS

2=Maintain definition      8=Edit reference authority

Type      Description                                Data Source      Creation      Authority Ref Obj
      Obj. File
_ CBLJDE JDE Cobol Program                        N JDESRC
_ CLPJDE JDE CL Program                          N JDESRC
_ CMD      Command                              N JDESRC
_ CPYCBL /COPY Cobol Member                      N JDESRC
_ CPYRPG /COPY RPG Member                       N JDESRC
_ DSPJDE JDE Display File                        N JDESRC
_ DTAARA Data Area                              Y
_ JDEDD Data Dictionary items                    N
_ JDEDW DREAM Writers                          N
_ JDEMNS Menus                                N
_ JDEMNU Menus                                N
_ JDEUDC User Defined Codes                    N
_ JDEVO Vocabulary Overrides                    N

More...

F3=Exit  F4=Add/Delete types  F6=Create missing items
F8=User-defined parameters  F12=Cancel
  
```

- 4 To omit or add types, press F4 to view the list of *all* type codes that are defined, and then re-select those you need.

The source file names for JD Edwards World IBM i object types default to JDESRC, which is the World standard. You can change these across all levels of your application by changing the global type code entries (on the main menu, select option 8, then option 4), or you can change them at each level of the application by typing 2 next to each level's type code entry and changing the source file names.

Use source file name JDFSRC/JDESRC for the permanently locked application level that defines your JD Edwards World base application.

- 5 Type 2 next to an item.

A panel like this appears:

MM/DD/YYYY		RPGJDE Defaults		Your Company, Inc.	
14:29:34	Application: 20JD	Rel:	Ver:	Lev: 1	SYSTEM: YOURSYS
Type choices and press Enter.					
				Field Attributes R=Restrict, L=Lock	
From Source file	JDESRC	Name			<u>L</u>
Library	DEVSRC	Name, *PGMR			<u>L</u>
Target Source file	JDESRC	Name			<u>L</u>
Library	TESTSRC	Name, *OBJLIB			<u>L</u>
From object library	DEVOB	Name, *PGMR			<u>L</u>
Target object library	TESTOB	Name			<u>L</u>
Naming mask					—
Delete test object	—	Y, N			—
Delete test source	—	Y, N			—
Method		CSCO, MO, CD, COPR			—
		CSMO, CSCD, CS, *NULL			—
F3=Exit	F12=Cancel				More...

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

Normally, the source file name is set to a default of JDESRC.

Note For applications that will use the JD Edwards World data-driven type codes, the **From Source Library** must match the **From object library**, and the **Target Source Library** must match the **Target object library**.

Also, for data-driven type codes, you cannot use *PROJECT to specify these library names. Instead, you must type the library name or use *PGMR.

The JD Edwards World object type codes issue TURNOVER® Lifecycle Manager-supplied create commands, which call TURNOVER® Lifecycle Manager programs. These programs call the appropriate JD Edwards World program, usually P98CRTPGM, depending on what JD Edwards World release you are running. Wherever appropriate, the P98CRTPGM program gets parameter values such as GENLVL and XREF from the World repository.

These create commands in TURNOVER® Lifecycle Manager to make it easier to understand the parameters required for each program, and to customize or create your own type entries. Also, the TURNOVER® Lifecycle Manager-supplied commands recognize whether the object was created successfully. This is essential for TURNOVER® Lifecycle Manager to determine whether to continue with the promotion process, or back out your changes. The source for the TURNOVER® Lifecycle Manager interface programs is in the TURNOVER® Lifecycle Manager product library.

More information

For information about **Use F98CRTCMD File**, **Use P98CRTPGM Program**, and other parameters for the TURNOVER® Lifecycle Manager-supplied command, see *Managing JD Edwards World components' on page 42*.

For information about other fields, press F1.

Setup wizard: TSETUPJDE command parameters

The TSETUPJDE command has one parameter, which identifies which component you want to configure.

Value	Description
*SELECT	Displays a panel appears with a list of components. You can press Enter to see them all in sequence, or type space next to any options you do not want to run. This is the default.
*OVERVIEW	Displays the Overview panels. This option is for information only; no verification or setup processing occurs.
*ENV	Analyzes and configures the TURNOVER® Lifecycle Manager environment library list.
*USEROPT	Detects and configures JD Edwards World user-defined options for the Programmer Worklist.
*APPL	Detects and configures TURNOVER® Lifecycle Manager application definitions for managing JD Edwards World objects.
*LIBLIST	Verifies and adjusts the application library lists for your JD Edwards World applications.
*TYPECODES	Verifies and corrects the application type code list specified for your JD Edwards World applications.
*JDEFFILES	Verifies (for each JD Edwards World application that manages data-driven objects) that the application target libraries contain the World definition files for the associated data-driven objects. If the files do not exist and you opt to create them, they are copied from the environment's JD Edwards World data library.
*SVR	Checks to see if the JD Edwards applications in your environment are configured to permit automatic SVR updates.
*TURCRARE	Displays a panel that reminds you to check the TURCRARE job descriptions on your remote systems. The TURCRARE job descriptions must also include the appropriate JD Edwards World libraries in their library lists. This option is only for information; no verification or setup processing occurs.
*USEALTENV	Verifies that alternate environment configurations have been defined and enables them for TURNOVER® Lifecycle Manager's use.

3 Managing JD Edwards World components

See

'Managing JD Edwards World native object types' on page 42

'Creating JD Edwards World programs with embedded SQL' on page 46

'Creating additional JD Edwards World object types' on page 47

'Specifying the target release' on page 48

'Retrieving compile options from the SVR' on page 49

'Using an exit point to update the SVR' on page 50

Managing JD Edwards World native object types

Each of the JD Edwards World native types (CBLJDE, CLJDE, and so on) is defined in the TURNOVER® Lifecycle Manager type code file. These World definitions are shipped with TURNOVER® Lifecycle Manager and require no special installation steps.

To view the TURNOVER® Lifecycle Manager type codes

- 1 On the TURNOVER® Lifecycle Manager main menu, select option 8 (Utility menu), and then option 4 (Work with type code definitions).
- 2 On the Work with Type Codes panel, type 2 next to a type code, and then press Enter.

The example panel below shows the type code that TURNOVER® Lifecycle Manager uses when you check out and promote JD Edwards World RPG programs. Other program object types are defined in a similar way.

MM/DD/YYYY 14:14:44	Change a Type Code	Your Company, Inc. YOURSYS																																				
Type code	RPGJDE	CM scheme *TURNOVER																																				
Description	JDE RPG Program	Attribute RPG																																				
Object type	*PGM	Source file name . . JDESRC																																				
Sequence	500	Data object N																																				
Create command . . .	SOFTTURNE/JDERPG MEMBER("&OB") FILE("&SF") OBJLIB("&LI") SRCLIB("&SL")																																					
<table border="0"> <tr> <td>"&OB" = Object name</td> <td>"&RF" = Reference</td> <td>"&X0"-"&X9" = Exit</td> </tr> <tr> <td>"&LI" = Library name</td> <td>"&TO" = Test object name</td> <td></td> </tr> <tr> <td>"&TY" = Type code</td> <td>"&TL" = Test object library name</td> <td></td> </tr> <tr> <td>"&SM" = Source member name</td> <td>"&TR" = Target release</td> <td></td> </tr> <tr> <td>"&SF" = Source file name</td> <td>"&FM" = From source member name</td> <td></td> </tr> <tr> <td>"&SL" = Source library name</td> <td>"&FF" = From source file name</td> <td></td> </tr> <tr> <td>"&U0" = Generation options</td> <td>"&FL" = From source library name</td> <td></td> </tr> <tr> <td>"&U2" = Optimization level</td> <td>"&U1" = Debugging views</td> <td></td> </tr> <tr> <td>"&U4" = Source listing options</td> <td>"&U3" = Compiler options</td> <td></td> </tr> <tr> <td>"&U6" = Unassigned</td> <td>"&U5" = Unassigned</td> <td></td> </tr> <tr> <td>"&U8" = Binding directory</td> <td>"&U7" = Service program source file</td> <td></td> </tr> <tr> <td>F3=Exit F4=Prompt/select F7=Applications F12=Cancel F22=Long command</td> <td>"&U9" = Module source file</td> <td></td> </tr> </table>			"&OB" = Object name	"&RF" = Reference	"&X0"-"&X9" = Exit	"&LI" = Library name	"&TO" = Test object name		"&TY" = Type code	"&TL" = Test object library name		"&SM" = Source member name	"&TR" = Target release		"&SF" = Source file name	"&FM" = From source member name		"&SL" = Source library name	"&FF" = From source file name		"&U0" = Generation options	"&FL" = From source library name		"&U2" = Optimization level	"&U1" = Debugging views		"&U4" = Source listing options	"&U3" = Compiler options		"&U6" = Unassigned	"&U5" = Unassigned		"&U8" = Binding directory	"&U7" = Service program source file		F3=Exit F4=Prompt/select F7=Applications F12=Cancel F22=Long command	"&U9" = Module source file	
"&OB" = Object name	"&RF" = Reference	"&X0"-"&X9" = Exit																																				
"&LI" = Library name	"&TO" = Test object name																																					
"&TY" = Type code	"&TL" = Test object library name																																					
"&SM" = Source member name	"&TR" = Target release																																					
"&SF" = Source file name	"&FM" = From source member name																																					
"&SL" = Source library name	"&FF" = From source file name																																					
"&U0" = Generation options	"&FL" = From source library name																																					
"&U2" = Optimization level	"&U1" = Debugging views																																					
"&U4" = Source listing options	"&U3" = Compiler options																																					
"&U6" = Unassigned	"&U5" = Unassigned																																					
"&U8" = Binding directory	"&U7" = Service program source file																																					
F3=Exit F4=Prompt/select F7=Applications F12=Cancel F22=Long command	"&U9" = Module source file																																					

- Go to the **Create command** line, and then press F4 to see all of the command parameters. See *'Create JD Edwards RPG Program command: JDERPG' on page 43.*

Create JD Edwards RPG Program command: JDERPG

Create RPG Program (JDERPG)		
Type choices, press Enter.		
Program	> "&OB"	PGM
File ID	> "&SF"	SRCFILE
Print Option	1	0, 1, 2, 3, 4
Object Lib	> "&LI"	LIB
Source Lib	> "&SL"	LIB
Create Option	1	OPT
Cross Reference (Y/N)	N	XREF
Generation Level	—	GENLVL
Use F98CRTCMD File	Y	Y, N
Use P98CRTPGM Program	Y	Y, N
Function code	RPG	RPG, RPGL
Bottom		
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display F24=More keys		

You want to override any parameter values.

The substitution variables such as &OB are handled automatically by TURNOVER® Lifecycle Manager at run time by substituting the variable with the object names, library names, and so on, that are on the TURNOVER® Lifecycle Manager form line representing the item being promoted.

Program (MEMBER)

The source file member name for the program you are promoting.

File ID (FILE)

The source file that contains the source member for the program you are promoting.

Print Option (PRINTOPT)

Value	Description
1	Controls (for OPM programs) the Source listing options and Generation options parameters for the CRTRPGPGM and CRTRPTPGM commands by setting these parameters to OPTION (*SOURCE) GENOPT (*NOLIST). This value has no effect on the creation of ILE programs.
2	Overrides the print file compile listing to HOLD (*YES) and also sets the parameters as described for PRINTOPT (1).
3	Sets the Generation options parameter for the CRTRPGPGM and CRTRPTPGM commands to GENOPT (*NOGEN) and holds the output on the output queue. Only this value affects SQL programs (either OPM or ILE).
4	Produces the same results as PRINTOPT (1), except that JD Edwards World deletes the spooled file that is created from the compile.
any other value	Sets the Source listing options and Generation options parameters for the CRTRPGPGM and CRTRPTPGM commands to OPTION (*NOSRC) GENOPT (*NOLIST)

Object Lib (OBJLIB)

The library into which you want TURNOVER® Lifecycle Manager to create the object.

Source Lib (SRCLIB)

The Library in which the source file (FILE) resides.

Create Option (CRTOPT)

Specify what kind of program you want TURNOVER® Lifecycle Manager to create when you set the **Use P98CRTPGM Program** parameter (see below) to N. The values for OPM programs (except CRTOPT(S)) are:

Value	Description
0	Issues the Create Auto Report RPG Program (CRTRPTPGM) command: for OPM programs.
1	Issues the Create RPG/400 Program (CRTRPGPGM) command: for OPM programs.
S	Issues the Create SQL RPG Program (CRTSQLRPG) command: for OPM programs (when the FUNC parameter is set to RPG).

Value	Description
-------	-------------

S	Issues the Create SQL ILE RPG Object (CRTSQLRPGI) command: for ILE programs (when the FUNC parameter is set to RPGL).
---	---

If you set **Use P98CRTPGM Program** to Y, TURNOVER® Lifecycle Manager uses the SVR to determine these values, and it ignores any value you enter for this parameter. However, TURNOVER® Lifecycle Manager passes your value to the command exit program, if it is active. For information about the exit program, see *'Using an exit point to update the SVR' on page 50*.

Cross Reference (XREF)

Type Y if you want to compile RPG programs with **Source listing options** set to `OPTION (*XREF)` or ILE programs with **Compiler options** set to `OPTION (*REF)`.

For more information, see the **Source listing options** parameter of the `CRTTRPGPGM` command or the **Compile options** parameter of the `CRTBNDRPG` command.

Generation Level (GENL)

(Applies only if **Use P98CRTPGM Program** is set to Y.)

The **Generation Severity Level** parameter to be specified when creating the program. For more information, see the **Generation severity level** parameter of the `CRTTRPGPGM` or `CRTBNDRPG` command.

A common mistake made is setting the GENLVL parameter to a value greater than 20 when trying to create an RPGLE program. The `CRTBNDRPG` command parameter GENLVL only allows values 0 – 20.

Use F98CRTCMD File (F98CRTCMD)

This is a source physical file in the JD Edwards World applications that *may* contain a member corresponding to the object name you are creating. If this file is present, it contains CL statements that the `JDERPG` command should use as overrides when creating the object. These may include the override database file (`OVRDBF`) command.

Set this to N if you will set the value of the **Use P98CRTPGM Program** parameter (see next) to Y. However, you can set both the **Use F98CRTCMD File** and **Use P98CRTPGM Program** parameters to Y without adverse consequences. The default, as shipped, is N.

If you are using a version of World that issues the commands in the `F98CRTCMD` file automatically, set this parameter to N.

Use P98CRTPGM Program (P98CRTPGM)

Value	Description
Y	TURNOVER® Lifecycle Manager calls the P98CRTPGM program, which gets certain values from the JD Edwards World SVR and passes them to the J98CRTRPG or J98CRTRPGL program. Other parameters are set based on the values you specify for this JDERPG command. Review the P98CRTPGM program source to determine what values this program gets, and under what circumstances.
N	TURNOVER® Lifecycle Manager directly calls the J98CRTRPG or J98CRTRPGL program (which of these programs TURNOVER® Lifecycle Manager calls depends on the value you specify for the Function code parameter; see next), using the values you specify for this JDERPG command.

Function code (FUNC)

(Applies only if **Use P98CRTPGM Program** is set to Y.)

The type of object you are defining.

Value	Description
RPG	TURNOVER® Lifecycle Manager calls the J98CRTRPG program for OPM programs.
RPGL	TURNOVER® Lifecycle Manager calls the J98CRTRPGL program for ILE programs.

If you set **Use P98CRTPGM Program** to Y, P98CRTPGM calls the OPM and ILE programs. Otherwise, TURNOVER® Lifecycle Manager calls them directly.

Creating JD Edwards World programs with embedded SQL

The TURNOVER® Lifecycle Manager - JD Edwards World interface updates the JD Edwards World SVR when you check out and promote IBM i objects. However, when SVR updates occur, sometimes (and SQL programs are one of these cases) not enough information is available to update every field.

To promote JD Edwards World programs that contain imbedded SQL

- 1 Add the program object to your worklist using an appropriately defined type code.
 For information about setting up type codes, see *'JD Edwards World IBM i object type codes and create commands' on page 7* and *'Creating additional JD Edwards World object types' on page 47*.
- 2 Check out the source to place a copy of it in your development library.
- 3 Edit the source as you normally would.

However, you might have to change the member attribute. You do not need to do this for existing members with the correct attribute. TURNOVER® Lifecycle Manager uses the existing member attribute. However, for new members, press F13 in SEU to change **Source type** to something like SQLRPG or SQLRPGLE.

- 4 Save the source, and then compile it.
- 5 Set specific JDERPG command parameters as follows:
 - Set **Create Option** to S.
 - Set **Use P98CRTPGM Program** to N.
 - Set **Function code (FUNC)** to RPG or RPGL (as is necessary).

If you use the P98CRTPGM program to get values from the SVR, you must update the SVR to set **Maint/RSTDSP** to S before you compile. You can set this value manually from the SVR, or use the TJDECMDEXT exit point to update the SVR. See ['Using an exit point to update the SVR' on page 50](#).

To create a type code for SQL objects

To make things easier, create a new type code, with the correct overridden parameters, which you use to manage only SQL objects. Define the type code as follows:

Property	Value
Description	JDE SQLRPGLE Programs
Object type	*PGM
Sequence	500
CM scheme	*TURNOVER
Attribute	JDERPGL
Source file name	JDESRC
Data object	N
Create command	SOFTTURNE/JDERPG MEMBER("&OB") FILE("&SF") OBJLIB("&LI") SRCLIB("&SL") CRTOPT(S) XREF(N) P98CRTPGM(N) FUNC(RPGL)

Creating additional JD Edwards World object types

To create additional customized JD Edwards World type codes for native types, copy the type code that corresponds to the object type you need, and then change the **Attribute** parameter.

By creating additional type codes in this way, you ensure that the SVR gets updated correctly.

TURNOVER® Lifecycle Manager supports automatic updating of the SVR of additional JD Edwards World IBM i object types such as /COPY types.

To update the SVR with the object information automatically

- 1 Create the type code if you need to.
- 2 Set the **Attribute** parameter as JDExxxx (where xxxx is a code defined in the JDE UDC table 98 FN). The **Attribute** parameter must begin with JDE, followed by type of object you want to create (CLP, RPG, CBL, DSPF, RPGL, RPGS, and so on), for example: JDECLP.

/COPY types

For /COPY types, use the **File prefix** parameter for the ADDPWLITM (Add Programmer Worklist Item) command to help you define what the SVR “base member” should be.

Add Programmer Worklist Item (ADDPWLITM)		
Type choices, press Enter.		
List	> 'JDE 0001'	Name, *SELECT, *ACTIVE
Object	> C00311	Character value, *FORM
Attribute	> JDCOPY	*SELECT, *ALL, AAATEST, AAU...
Checkout item	*NO	*NO, *YES
Description	*OBJD	
Programmer	*LIST	Name, *LIST
Reference	*LIST	*LIST, Char
User defined status	NEW	Character value
Application	*LIST	Character value, *LIST
Release		Number
Version		Number
Create worklist	*NO	*NO, *YES
Form number	*NONE	Number, *NONE
File prefix		Character value
Function use	*SELECT	Character value, *SELECT...
Bottom		
F3=Exit	F4=Prompt	F5=Refresh
F13=How to use this display	F10=Additional parameters	F12=Cancel
	F24=More keys	

Specify any character in **File prefix**. When TURNOVER® Lifecycle Manager writes the record to the SVR, it replaces the first character of the object name with this prefix.

If you leave **File prefix** blank, TURNOVER® Lifecycle Manager gives the base member the same name as the /COPY member.

Specifying the target release

When you compile a JD Edwards World IBM i program, the World compiler program gets, the version of IBM i on which the program needs to run from your CASE profile, F98009 (or, in the absence of that, from the *PUBLIC CASE profile).

TURNOVER® Lifecycle Manager users can each have their own CASE profile in which the target release is specified. Also, every level of a TURNOVER® Lifecycle Manager application and every remote system to which you are distributing can have their own CASE profile file (F98009).

For each F98009 file, verify all users' CASE profiles, and the *PUBLIC CASE profile to ensure that your programs will be compiled at the correct target release.

For COBOL and CL programs

For COBOL and CL programs only, the TURNOVER® Lifecycle Manager commands JDECBL (Create Cobol Program) and JDECLP (Create CL Program) allow you to specify a target release.

Create Cobol Program (JDECBL)

Type choices, press Enter.

Program	>	"&OB"	PGM
File ID		QLBLSRC	SRCFILE
Print Option		1	0, 1, 2, 3, 4
Object Lib			LIB
Source Lib			LIB
Generation Level			GENLVL
Cross Reference (Y/N)		N	XREF
Target Release		*CURRENT	TARGET
Use F98CRTCMD File		Y	Y, N
Use F98CRTPGM Program		Y	Y, N

Bottom

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

TURNOVER® Lifecycle Manager uses the value you specify for the **Target Release** parameter only if you set **Use P98CRTPGM Program** to N.

If you set the **Use P98CRTPGM Program** to Y, TURNOVER® Lifecycle Manager gets the target release from the CASE profile.

Retrieving compile options from the SVR

When you compile JD Edwards World native types, you occasionally need to compile these objects with values defined in the SVR. These values include generation level, maintenance option, form length, lines per inch, and so on.

The first time you compile an object that needs values from the SVR, you can get the values by using option 36 (Compile) on the worklist and then pressing F4.

The command's prompt override program then accesses the SVR to retrieve the necessary parameters. When you press Enter, TURNOVER® Lifecycle Manager remembers that these values have been changed for this object. When you add the object to a form, or the next time you compile the object, TURNOVER® Lifecycle Manager prompts you to use the changed values.

Using an exit point to update the SVR

When you override a create command, TURNOVER® Lifecycle Manager remembers your changes and prompts you to use that overridden creation command the next time you promote the object through TURNOVER® Lifecycle Manager.

However, you might want to also update the JD Edwards World SVR with the overridden values. You can do this by using an exit point. The TURNOVER® Lifecycle Manager commands from which you can use this exit point are:

JDECBL
 JDECLP
 JDEDSPF
 JDELF
 JDEPF
 JDEPRTF
 JDERPG

Exit example

Suppose you want to change a print file to use a different form type. Prompt the JDEPRTF command, and then change the form type value. You can then use the exit point from the JDEPRTF command, so that when you ran that command, TURNOVER® Lifecycle Manager calls an exit program to update records in, or write records to, the JD Edwards World SVR file, F9805.

Creating an exit program

To use the exit point, create a program that accepts these two variables:

COMMANDS

(*CHAR 10) The value passed will be the name of the TURNOVER® Lifecycle Manager JD Edwards World creation command that issued the call to your exit program.

DATA

(*CHAR 100) The data passed varies depending upon which command issued the call. All parameters used in the TURNOVER® Lifecycle Manager JD Edwards World creation commands are passed, in order, in this variable. Sample data structures are available in the source member TJDECMDEXT in source file ARPGLESRC in your TURNOVER® Lifecycle Manager product library.

Defining the exit program to TURNOVER® Lifecycle Manager

To define your exit program to TURNOVER® Lifecycle Manager, create a data area named TJDECMDEXT (*CHAR 20) in the TURNOVER® Lifecycle Manager product library (usually SOFTTURN).

- The first 10 positions of this data area hold the name of the exit program you want TURNOVER® Lifecycle Manager to call.

- The second 10 positions hold the library name where the program resides. If while executing one of the TURNOVER® Lifecycle Manager commands listed above, TURNOVER® Lifecycle Manager finds this program in the library specified, it calls that program.

Note

Test your exit program. The library list in effect when you issue these commands might vary, depending on how the command was issued and various other TURNOVER® Lifecycle Manager configurations. These commands have your TURNOVER® Lifecycle Manager product library specified as the PRODUCT library of the command. User libraries might be your current library list, or the application library list. Because of these factors, you might want to verify which SVR files you are updating. Do not change the library list in your program: if you do change it, reset it to the way it was before you changed it.

4 Managing Dream Writer, World Writer, and FASTR versions

TURNOVER® Lifecycle Manager for Dream Writers, and TURNOVER® Lifecycle Manager for World Writers and FASTRs enable you to manage single versions, or a range of versions.

Setting up versions

- 1 On the TURNOVER® Lifecycle Manager main menu, choose option 8 (Utility menu), and then option 4 (Work with type code definitions).
- 2 Press F6 to define the type codes JDEDWV, JDEWWV, and JDEFSV.

The Add a Type Code panel appears.

MM/DD/YYYY	Add a Type Code	Your Company, Inc.
10:20:02		YOURSYS
Type code	JDEDWV	CM scheme
Description	Dream Writer Versions	Attribute
Object type	*JDEDWV	Source file name
Sequence	100	Data object
Create command	SOFTTURN/CPYJDEOBJ OBJ('&SM') OBJTYPE(*JDEDWV) FROMLIB(' &FL') TOLIB('&SL') CPYOPT(*REPLACE)	
"&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 F8=Add IFS type code F12=Cancel F22=Long command		

Attribute

The value in the Attribute field must match the type code name.

Object type

The value in the **Object type** field (as well as the OBJTYPE parameter of the command) must be *JDEDWV, *JDEWWV, or *JDEFSV.

Data object

If you need the versions to explode to another library upon promotion, set **Data object** to Y.

Create command

The create command is CPYJDEOBJ. Be sure to specify your TURNOVER® Lifecycle Manager product library.

The OBJ parameter value has single quotes around the double quotes.

- 3 Press Enter.
- 4 Add this type code to each level of all the applications you use to manage versions.

Note While most soft-coded types are held in the JD Edwards World common library, World Writers might be held in the JD Edwards World data library. Therefore, when adding the World Writer type code to your application, make sure that you specify the correct library based on your JD Edwards setup.

Using version support

- 1 In your Programmer Worklist, press F6 to add an item.

The Add Programmer Worklist Item panel appears.

Add Programmer Worklist Item (ADDPWLITM)		
Type choices, press Enter.		
List	> 'JDE 0001'	Name, *SELECT, *ACTIVE
Object	_____	Character value, *FORM
Attribute	_____	*SELECT, *ALL, AAATEST, AAU...
Checkout item	*NO	*NO, *YES
Description	*OBJD	
Programmer	*LIST	Name, *LIST
Reference	*LIST	*LIST, Char
User defined status	NEW	Character value
Application	*LIST	Character value, *LIST
Release	_____	Number
Version	_____	Number
Create worklist	*NO	*NO, *YES
Form number	*NONE	Number, *NONE
Bottom		
F3=Exit	F4=Prompt	F5=Refresh
F13=How to use this display	F10=Additional parameters	F12=Cancel
	F24=More keys	

- 2 In the **Object** field, specify the Dream Writer form ID, World Writer group, or FASTR form ID.
- 3 In the **Attribute** field, specify the appropriate type code.
- 4 Fill in any other necessary parameters, and then press Enter.

The **From version** and **To version** fields appear.

Add Programmer Worklist Item (ADDPWLITM)		
Type choices, press Enter.		
List	> 'JDE 0001'	Name, *SELECT, *ACTIVE
Object	> P00311	Character value, *FORM
Attribute	> JDEDWV	*SELECT, *ALL, AAATEST, AAU...
Checkout item	*NO	*NO, *YES
Description	*OBJD	
Programmer	*LIST	Name, *LIST
Reference	*LIST	*LIST, Char
User defined status	NEW	Character value
Application	*LIST	Character value, *LIST
Release		Number
Version		Number
Create worklist	*NO	*NO, *YES
Form number	*NONE	Number, *NONE
From version		Name, generic*
To version		Character value
Bottom		
F3=Exit F4=Prompt F5=Refresh F10=Additional parameters F12=Cancel		
F13=How to use this display F24=More keys		

5 Specify the version range you want to check out.

- To check out a single version, type the same value in each field.
- To check out all versions, leave both fields blank.
- To check out all versions that begin with the same characters, use the * (asterisk) wildcard character in the **From version** field and leave the **To version** field blank.

For example, to check out all versions that begin with the characters TEST, type TEST* in the **From version** field and leave the **To version** field blank.

6 When the object is on your worklist, you can work on it as you do any other object. For example, use option 32 to edit it, option 46 to add it to a form, and so on.

What happens under the covers

In the worklist, it might appear that these objects are not in alphabetical order.

The IBM i structure has libraries, objects, and object types. Each library can only have one object of a particular name of any given type. Therefore, many of the TURNOVER® Lifecycle Manager files have keys over these items. However, in managing versions, there are two additional fields, **From version** and **To version**. To account for this, the object type, the object name (Form ID or Group), and the version information are associated with an object alias.

For example, suppose your Dream Writer P01053 has versions AB0001 through AB0010, and that you first want to change versions AB0008 and AB0009. You will add P01053 to your worklist, specifying AB0008 as the **From version** and AB0009 as the **To version**. This translates to the following alias:

*000000001 = P01053 *JDEDWV AB0008 AB0009

What you see on the panel is P01053, but what it actually represents is *000000001. (TURNOVER® Lifecycle Manager automatically generates the alias name. The *000000001 alias, and the other aliases shown in these examples, are for illustration only. Whenever *000000001 is slated to appear on the panel, it is translated it to P01053.

Next, you might want to change P01053 versions AB0003 through AB0006. You will add P01053 to your worklist and specify AB0003 as the **From version** and AB0006 as the **To version**. This translates to the following alias:

```
*0000000002 = P01053 *JDEDWV AB0003 AB0006
```

A third change might be for Dream Writer P01051, versions AB0001 through AB0010, translated as follows:

```
*0000000003 = P01051 *JDEDWV AB0001 AB0010
```

Now, when these objects appear on the worklist, or even in object history, they are in the order of their alias (*000000001, *000000002, *000000003), instead of the order you might expect (P01051/AB0001-AB0010, P01053/AB0003-AB0006, P01053/AB0008-AB0009).

Interface changes for version support

To find these objects once they are ordered by their alias, you can use F4 to prompt on various **Position to** fields.

To position your worklist to a particular object that was added to the worklist as one of these special types, move the cursor to **Position to**, and then press F4.

A list appears of all Dream Writers, World Writers, or FASTRs on the current worklist that were added with these type codes.

MM/DD/YYYY 10:42:24		Work with Programmer Worklist		Your Company, Inc.	
				YOURSYS	
Worklist JDE 0079		JDEWWV and JDEFSV			
Position to		(Object)		Apply filters Y (F17=Filters)	
Appl/Rel/Ver . . .				e)	
2=Change 3=Copy		Position to form ID			
21=Checkout				flicts	
		Worklist . . JDE 0079		Nxt	
Object At		Position to:		Filter type: _____	
___ P75301 JD		1=Select		Opt	
___ P75301 JD					
___ P75301 JD		Form ID Type		From version To version 32	
___ Q08 JD		CHINAFIN *JDEFSV		9999999999	
___ Q08 JD		CHINAFIN *JDEFSV DOUG		DOUG9999999	
___ Q08 JD		P01051 *JDEDWV AB0001		AB0010 32	
___ CHINAFIN JD		P01053 *JDEDWV AB0003		AB0006	
___ CHINAFIN JD		P01053 *JDEDWV AB0008		AB0009 46	
___ CHINAFIN JD		P75301 *JDEDWV		9999999999 32	
___ P75301		*JDEDWV DOUG		DOUG9999999 re...	
Parameters or com		Q08 *JDEWWV		9999999999	
==> _____				More... _____	
F2=Forms F3=Exit					
F10=Command entry		F3=Exit F12=Cancel		keys	

The panel displays the objects in a more logical order. Type 1 next to the object and version range you want to position to, and then press Enter.

To see which range you are working with on the worklist, press F11 until you see the **Source** information. Your **From** and **To** versions are listed under the **Source Library** and **Source File** fields.

When viewing a form line for these objects, you now see a different view that is tailored to these types.

Checking out versions

You cannot check out overlapping versions for the same object. The checkout flag on the worklist shows a P (for Partial), indicating that part of the specified version range is already checked out.

- When you check out an object using the JDEDW type code, you do not have to check out that Dream Writer's Processing Options. Processing Options are part of the Dream Writer; they are promoted with it when you promote the Dream Writer using the JDEDW type code.
- When you promote the Dream Writer using a JDEDWV type code, the Processing Options are not promoted with the Dream Writer (unless they do not exist in the target library).

Therefore, when you are creating new Dream Writers, you should use the JDEDW type code and, when modifying or adding versions to an existing Dream Writer, you should use the JDEDWV type code.

Use the JDEPO type code to modify the Processing Options setup for existing Dream Writers. Use this type code with JDEDWV, or alone. Do not use the JDEPO type code with JDEDW.

Testing the JDEDWV, JDEWWV, and JDEFSV type codes

Before you implement the type codes in a live environment, test them to make sure that they are working as you expected.

When you are satisfied that the new type codes are working as you expect, you might want to remove the old type codes (JDEDW, JWW*, JFS*) from your application definitions to prevent further use of them.

Before you remove the old type codes, make sure that you complete all work in progress that uses them.

More about JD Edwards World vocabulary overrides (JDEVO)

This section applies to JD Edwards Release 6.2 with PTF 00005, and later.

JD Edwards Release 6.2 includes new parameters for copy programs to better manage Vocabulary Overrides. These parameters are language (2 positions) and system (4 positions). To manage Vocabulary Overrides, you must understand how TURNOVER® Lifecycle Manager determines these parameter values.

Type Code file

Consider first the TURNOVER® Lifecycle Manager type code.

You must have an entry in the file for each combination of language and system you use. If you use more than one, name them JDEVO_x, where *x* is any character (example: JDEVOF, for French). The Object type parameter used should be *JVOLLSSSS, where *ll* = language and *ssss* = system.

MM/DD/YYYY	Change a Type Code	Your Company, Inc.
15:49:48		YOURSYS
Type code	JDEVO	CM scheme *TURNOVER
Description	<u>Vocabulary Overrides</u>	Attribute <u>JDEVO</u>
Object type	<u>*JVOFR0021</u>	Source file name
Sequence	<u>100</u>	Data object <u>N</u>
Create command	<u>SOFTTURNE/CPYJDEOBJ OBJ("&SM") OBJTYPE(*JVOFR0021) FROMLIB</u> <u>("&FL") TOLIB("&SL") CPYOPT(*REPLACE)</u>	
<u>"&OB"</u> = Object name	"&RF" = Reference	"&X0"-"&X9" = Exit
<u>"&LI"</u> = Library name	"&TO" = Test object name	
<u>"&TY"</u> = Type code	"&TL" = Test object library name	
<u>"&SM"</u> = Source member name	"&TR" = Target release	
<u>"&SF"</u> = Source file name	"&FM" = From source member name	
<u>"&SL"</u> = Source library name	"&FF" = From source file name	
<u>"&U0"</u> = Generation options	"&FL" = From source library name	
<u>"&U2"</u> = Optimization level	"&U1" = Debugging views	
<u>"&U4"</u> = Source listing options	"&U3" = Compiler options	
<u>"&U6"</u> = Unassigned	"&U5" = Unassigned	
<u>"&U8"</u> = Binding directory	"&U7" = Service program source file	
F3=Exit F4=Prompt/select F7=Applications F12=Cancel F22=Long command	"&U9" = Module source file	

To define more than one TURNOVER® Lifecycle Manager type code, you can copy the first one you create and change the language and system code.

Application definition

You must also include the TURNOVER® Lifecycle Manager type code in your TURNOVER® Lifecycle Manager application definition.

Define the From source file and Target source file names as *llssss*, where *ll* is the language, and *ssss* is the system.

How vocabulary overrides are promoted

For all *JDEVO types, TURNOVER® Lifecycle Manager's JD Edwards World copy program (CPYJDEOBJ as defined in the TURNOVER® Lifecycle Manager type code entry) calls program P996302MM, if it is in the library list, or program P996302, if P996302MM is not found. (P996302MM is an early version of the program, used at some World customer sites.)

TURNOVER® Lifecycle Manager passes the language, system parameters, and all of the other parameters that the P996302 program requires to copy the VO entries from one environment to another.

Obsolete type codes for World Writers and FASTRs

TURNOVER® Lifecycle Manager supports World Writers and FASTRs with the new type codes JDEWWV and JDEFSV.

These new type codes require less maintenance than the previous JWW and JFS type codes. The old type codes required you to create multiple type codes for each individual World Writer group or FASTR form ID, and they only allowed the checkout and promotion of one version at a time.

The new type codes allow you to use only one type code for each World Writer group or FASTR form ID, and to check out and promote multiple versions with each.

If you are using the old type codes, switch to the new type codes. To do so, finish any work in progress. Remove the old type codes from your application, and then add the new type codes.

5 JD Edwards World data-driven object types

Each JD Edwards World data-driven object type uses the same create command: CPYJDEOBJ. The values for **Object type** control which data-driven type gets promoted. In this example, the type is *JDEDD (Data Dictionary item).

MM/DD/YYYY	Change a Type Code	Your Company, Inc.
14:47:21		YOURSYS
Type code	JDEDD	CM scheme *TURNOVER
Description	Data Dictionary items	Attribute JDEDD
Object type	*JDEDD	Source file name
Sequence	100	Data object N
Create command	SOFTTURN/CPYJDEOBJ OBJ("&SM") OBJTYPE(*JDEDD) FROMLIB("&FL") TOLIB("&SL") CPYOPT(*REPLACE)	
"&OB" = Object name "&LI" = Library name "&TY" = Type code "&SM" = Source member name "&SF" = Source file name "&SL" = Source library name "&U0" = Generation options "&U2" = Optimization level "&U4" = Source listing options "&U6" = Unassigned "&U8" = Binding directory F3=Exit F4=Prompt/select F7=Applications F12=Cancel F22=Long command "&RF" = Reference "&TO" = Test object name "&TL" = Test object library name "&TR" = Target release "&FM" = From source member name "&FF" = From source file name "&FL" = From source library name "&U1" = Debugging views "&U3" = Compiler options "&U5" = Unassigned "&U7" = Service program source file "&U9" = Module source file		

TURNOVER® Lifecycle Manager handles any substitution variables such as &OB at runtime by substituting the variable with the object names, library names, and so on, from the TURNOVER® Lifecycle Manager form line for the item being promoted. To see the command panel, move the cursor to the **Create command** line, and then press F4.

Copy JDE Object command (CPYJDEOBJ)

Copy JDE Object (CPYJDEOBJ)

Type choices, press Enter.

Object	> "&SM"	Character value
Object type	> *JDEDD	Character value
From library	> "&FL"	Character value
To library	> "&SL"	Character value
Copy option	> *REPLACE	*REPLACE, *ADD
Create files	*NO	*NO, *YES

Bottom

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

Object

The name of the data-driven object.

Object type

One of these data-driven object types:

Object type	Description
*JDEDD	Data Dictionary item
*JDD	
*JDEDW	Dream Writer
*JDEDWV	
*JDW	
*JDEFSV	Fasters
*JDEPO	Processing options
*JDEUDC	User-defined codes
*JUD	
*JDEVO	Vocabulary override
*JVO	
*JDEMNU	Menus
*JDEWWV	World Writer

From library

The library from which the item is being copied.

To library

The library to which the item is being copied.

Copy option

*ADD or *REPLACE.

For normal promotions, this is usually set to *REPLACE.

If you use this command elsewhere (for example, during checkout), you can use *ADD to avoid accidentally overlaying items in a test environment.

Create files

Normally, because the files containing the data-driven type definitions already exist in the target library, this is set to *NO during promotions.

However, you can set it to *YES if you use it outside of TURNOVER® Lifecycle Manager to create the JD Edwards World definition files in a test or development library.

Exploding JD Edwards World data-driven object types

TURNOVER® Lifecycle Manager allows you to copy (explode) the JD Edwards World data-driven types during a form promotion. If you need to have the World data-driven types exploded to another library during a form run, you must define this to TURNOVER® Lifecycle Manager, as follows:

Define the type code as a data object

To change global type code parameters, you must have TURNOVER® Lifecycle Manager system defaults authority.

- 1 On the TURNOVER® Lifecycle Manager main menu, select option 8 (Utility menu).
- 2 On the Utility menu, select option 4 (Work with type code definitions).
- 3 On the Work with Type Codes panel, specify Y in the **Data Object** column for the appropriate type code.

Specify the explode library in the application

- 1 On the TURNOVER® Lifecycle Manager main menu, select option 1 (Work with application definitions).
- 2 In the Work with Application Definitions panel, select the appropriate appl/rel/ver/level with option 11 (Library defaults).
- 3 In the Library Defaults panel, press F19 to explode the data libraries,

- 4 In the Maintain Data Explosion Table panel, press F6 to add the name of the library to which the data-driven object will be exploded.

TURNOVER® Lifecycle Manager by default explodes all data types to all libraries defined in the explode list. If you need to explode to only a certain set of these libraries, use Exit 8.

A form run for that *appl/rel/ver/level* causes TURNOVER® Lifecycle Manager to issue the CPYJDEOBJ command for each of the explode libraries listed in the Data Explosion Table. This copies the data-driven object from the applications target library into the explode libraries.

Language, system, and version support (for JDEDD, JDEDW, JDEWWV, JDEFSV, and JDEUDC)

TURNOVER® Lifecycle Manager has language and system support for JD Edwards World Data Dictionaries (JDEDD), language or version support for Dream Writers (JDEDW), version support for World Writers (JDEWWV) and FASTRs (JDEFSV), and language support only for User Defined Codes (JDEUDC):

- *Language support* enables you to specify a language parameter when checking out and promoting Data Dictionaries, Dream Writers, and User Defined Codes.
- *System support* enables you to specify a system parameter when checking out and promoting Data Dictionary versions.
- *Version support* for Dream Writers, World Writers, and FASTRs enables you to check out and promote a single version, or a range of versions. For more information, see [‘Managing Dream Writer, World Writer, and FASTR versions’ on page 52](#).

Setting up special object types

TURNOVER® Lifecycle Manager has special object types for language and system support. You can modify these samples and include them in your JD Edwards World applications as needed. These object types are:

```
*JDD11ssss  
*JDW11  
*JUD11
```

where *11* is language, and *ssss* is system.

These object types require specific setup in the TURNOVER® Lifecycle Manager Change a Type Code and application type code panels.

Example

This example uses a Data Dictionary with French language and system 9999:

MM/DD/YYYY 15:07:06	Change a Type Code	Your Company, Inc. YOURSYS																																				
Type code JDDFR9	CM scheme *TURNOVER																																					
Description JDEDD, French, 9999	Attribute JDDFR9																																					
Object type *JDDFR9999	Source file name FR9999																																					
Sequence 100	Data object N																																					
Create command SOFTTURN/CPYJDEOBJ OBJ("&SM") OBJTYPE(*JDDFR9999) FROMLIB("&FL") TOLIB("&SL") CPYOPT(*REPLACE)																																						
<table border="0"> <tr> <td>"&OB" = Object name</td> <td>"&RF" = Reference</td> <td>"&X0"-"&X9" = Exit</td> </tr> <tr> <td>"&LI" = Library name</td> <td>"&TO" = Test object name</td> <td></td> </tr> <tr> <td>"&TY" = Type code</td> <td>"&TL" = Test object library name</td> <td></td> </tr> <tr> <td>"&SM" = Source member name</td> <td>"&TR" = Target release</td> <td></td> </tr> <tr> <td>"&SF" = Source file name</td> <td>"&FM" = From source member name</td> <td></td> </tr> <tr> <td>"&SL" = Source library name</td> <td>"&FF" = From source file name</td> <td></td> </tr> <tr> <td>"&U0" = Generation options</td> <td>"&FL" = From source library name</td> <td></td> </tr> <tr> <td>"&U2" = Optimization level</td> <td>"&U1" = Debugging views</td> <td></td> </tr> <tr> <td>"&U4" = Source listing options</td> <td>"&U3" = Compiler options</td> <td></td> </tr> <tr> <td>"&U6" = Unassigned</td> <td>"&U5" = Unassigned</td> <td></td> </tr> <tr> <td>"&U8" = Binding directory</td> <td>"&U7" = Service program source file</td> <td></td> </tr> <tr> <td>F3=Exit F4=Prompt/select F7=Applications F12=Cancel F22=Long command</td> <td>"&U9" = Module source file</td> <td></td> </tr> </table>			"&OB" = Object name	"&RF" = Reference	"&X0"-"&X9" = Exit	"&LI" = Library name	"&TO" = Test object name		"&TY" = Type code	"&TL" = Test object library name		"&SM" = Source member name	"&TR" = Target release		"&SF" = Source file name	"&FM" = From source member name		"&SL" = Source library name	"&FF" = From source file name		"&U0" = Generation options	"&FL" = From source library name		"&U2" = Optimization level	"&U1" = Debugging views		"&U4" = Source listing options	"&U3" = Compiler options		"&U6" = Unassigned	"&U5" = Unassigned		"&U8" = Binding directory	"&U7" = Service program source file		F3=Exit F4=Prompt/select F7=Applications F12=Cancel F22=Long command	"&U9" = Module source file	
"&OB" = Object name	"&RF" = Reference	"&X0"-"&X9" = Exit																																				
"&LI" = Library name	"&TO" = Test object name																																					
"&TY" = Type code	"&TL" = Test object library name																																					
"&SM" = Source member name	"&TR" = Target release																																					
"&SF" = Source file name	"&FM" = From source member name																																					
"&SL" = Source library name	"&FF" = From source file name																																					
"&U0" = Generation options	"&FL" = From source library name																																					
"&U2" = Optimization level	"&U1" = Debugging views																																					
"&U4" = Source listing options	"&U3" = Compiler options																																					
"&U6" = Unassigned	"&U5" = Unassigned																																					
"&U8" = Binding directory	"&U7" = Service program source file																																					
F3=Exit F4=Prompt/select F7=Applications F12=Cancel F22=Long command	"&U9" = Module source file																																					

Properties

Type code

The type code can be anything, but it must start with the letter J.

Attribute

The Attribute must match the type code name.

Source file name

The Source file name must contain positions 5–10 of the object type.

Object type

The **Object type** must begin with *JDD, *JDW, or *JUD and include the appropriate language or system information, as follows:

- Positions 1-4 must contain *JDD, *JDW, or *JUD.
- Positions 5-6 must indicate the language.
- Positions 7-10 must indicate the system (only *JDD supports system).
- The language and system support for JD Edwards World data dictionaries, dream writers, and user-defined codes are shown in the following table.

Data-driven object type	Language support	System support
Data Dictionaries	*JDD11ssss	*JDD11ssss
Dream Writers	*JDW11	—
User Defined Codes	*JUD11	—

ll = language

ssss = system

- The command you use must be `CPYJDEOBJ`, and the `OBJTYPE` keyword in the create command must match the object type you specified.
- For Dream Writers (JDEDW), the requirements for checking out by language are:

The object type must be `*JDWll`.

The global source file name must be blank. (The blank name tells TURNOVER® Lifecycle Manager that checkout is to be by language.)

Setting up the type code in the application

MM/DD/YYYY	JDDFR9 Defaults	Your Company, Inc.	
10:29:34	Application: 20JD Rel: Ver: Lev: 1	YOURSYS	
Type choices and press Enter.			
		Field Attributes R=Restrict, L=Lock	
From Source file	FR9999	Name	<u>L</u>
Library	DEVCOM	Name, *PGMR	<u>L</u>
Target Source file	FR9999	Name	<u>L</u>
Library	TESTCOM	Name, *OBJLIB	<u>L</u>
From object library	DEVCOM	Name, *PGMR	<u>L</u>
Target object library	TESTCOM	Name	<u>L</u>
Naming mask	_____		—
Delete test object	_____	Y, N	—
Delete test source	_____	Y, N	—
Method	_____	CSCO, MO, CD, COPR	—
		CSMO, CSCD, CS, *NULL	
F3=Exit	F12=Cancel	More...	

The application type code panels all look similar to this one. The **From** and **Target** source file name, which is derived from positions 5-10 of the object type, must be the language or the language and system.

For applications that will use the JD Edwards World data-driven type codes, the **From Source Library** must match the **From object library**, and the **Target Source Library** must match the **Target object library**. For data-driven type codes, you cannot use `*PROJECT` to specify these library names; instead, you must type the library name or use `*PGMR`.

Language support for JD Edwards World soft-coded types

Various rules apply for maintaining *language variants* of JD Edwards World Vocabulary Overrides. These rules are:

- Create one type code with the object type of `*JVO` for the base language.

- Create additional JVOxxx type codes for each additional language, where xxx represents up to three characters you can use to identify the language (and, optionally, the system code). These type codes should have corresponding object types of *JVOxxxx. For example, you might have a JVOF type code with an object type of *JVOF (for French), and a JVOG type code with an object type of *JVOG (for German).
- Set the sequence number in the global type code for the variants to a higher number than the sequence number for the base language.

Checking out multiple *JVO types

To help you manage development of multiple language variants, you can check out multiple *JVOxxxx types for the same Vocabulary Override. TURNOVER® Lifecycle Manager also lets you check out *JDEPO and *JDW11 types at the same time. It does not, however, let you check out *JDEPO and *JDEDW types at the same time. Use *JDEPO to promote the base language and *JDW11 to promote the language variants.

Promoting language variants

When you promote (and check out), TURNOVER® Lifecycle Manager verifies that the base language for the Vocabulary Override is in place before promoting the language variants. If the base language is not present when you promote the language variant, TURNOVER® Lifecycle Manager will recover forms.

Editing types by language

When editing this type by language	TURNOVER® Lifecycle Manager calls
*JDW	P98OPTM3
*JDDxx	P9202

TURNOVER® Lifecycle Manager calls the P9201 program for *JDEDD types and *JDD base language types.

Promoting *JDD types

If a *JDD11 type is promoted, and JD Edwards World returns a code of 2 rather than end in error, the form ends with status of RAN-WRN. The TURNOVER® Lifecycle Manager job log shows the message, with details about which object and type had the warning.

(This is due to a possible World coding error: if either the F9202 or F9203 files do not contain records to copy, the return code is '2' even if the other of the two files had records to copy.)

Pseudo exit point

A "pseudo" exit point is available to help you manage deletion of soft-coded types in the target library. In most cases, you are not likely to use this exit point. However, if you do use it, you must create a program that accepts these parameters:

Parameter	Format
Form Number	*DEC 7,0
Line Number	*DEC 3,0
Object	*CHAR 10
Target Library	*CHAR 10
Type Code	*CHAR 10
Object type	*CHAR 10
Code	*CHAR 1 (A=Add, R=Replace, D=Delete)
DELETE	*CHAR 1 (Y=Yes, N=No; Y is the default)

Your program must accept the first seven parameters, and return a **Y** or **N** in the eighth parameter. If the program returns a **Y**, or if the form line is **D** (to delete), then TURNOVER® Lifecycle Manager deletes the soft-coded type in the target library after copying it to archive and before copying in the new (From) object.

To activate the exit point after the program has been compiled, create a data area called **TDLTJDEOBJ** (*CHAR 20). The first 10 positions of the data area must contain the program name; the second 10 positions must contain the name of the library where the program is.

DDG text and other unique JD Edwards World objects

You can create your own user-defined types to manage other unique objects such as Data Dictionary Glossary (DDG) text. For more information, see “Managing User-Defined Object Types” in the *TURNOVER® Lifecycle Manager Administrator’s Guide*.

6 Developing JD Edwards World objects with TURNOVER® Lifecycle Manager

Useful TURNOVER® Lifecycle Manager features

When managing any application, JD Edwards World or otherwise, there are several TURNOVER® Lifecycle Manager features you will find useful if you need to handle exceptional conditions.

Together, these features provide the flexibility you need to ensure that objects are created correctly, every time. Furthermore, these overrides are visible, so that any time you view TURNOVER® Lifecycle Manager's change history, you know exactly how the change was accomplished.

Pre-run commands

You can define pre-run commands (that is, commands that are included on a form to be run before your objects are promoted) to do overrides and other functions before creating an object. You can associate pre-run commands with any object on the form so that, when the object is changed in the future, TURNOVER® Lifecycle Manager remembers the override for you. Pre-run commands can apply to a single object (pre-line), or to all objects on a form (pre-form).

Create command overrides

When you override a create command parameter on a TURNOVER® Lifecycle Manager form, TURNOVER® Lifecycle Manager remembers the override and, the next time you promote the object, it prompts you to use the same override.

Create parameters

TURNOVER® Lifecycle Manager preserves important object parameters such as **Member size** for physical files and **User profile** for programs, by setting these parameters based on an existing object, on a reference object, or on the test version of the object being promoted.

User-defined create parameters

You can define default parameters that can be referenced by a create command and set to different values by application level.

For example, you can compile a program with GENOPT(*NOOPTIMIZE) when promoting into a test level and GENOPT(*OPTIMIZE) when promoting into production.

Object authority and ownership management

TURNOVER® Lifecycle Manager manages authority and object ownership by referring to the production object, a reference object or the test object being promoted.

Managing JD Edwards World objects from a programmer worklist

The TURNOVER® Lifecycle Manager Programmer Worklist Manager is a powerful development workbench panel that concentrates all of your program development activities on one panel. The Programmer Worklist is central to the development process.

- The Programmer Worklist helps you build a list of objects that need to be created or changed to fulfill any request or task.
- You can perform all of your program development tasks, quickly and efficiently.
- For version management, use the Related applications feature. You can view multiple versions of a Worklist item and make changes to all active versions of your software.
- Filters enable you to work with only the objects and environments you need, at any time.
- The X-Ref option enables you to view, checkout and/or compile related objects from TURNOVER® Lifecycle Manager's cross-reference files. Drill down to see all layers you need to be sure you included everything.
- You can customize the worklist by setting session defaults to control worklist sequence, test object creation, library lists, and so on.

All of the JD Edwards World IBM i and data-driven object types can be managed on the Programmer Worklist, along with any other IBM i object types. Checkout, editing, object creation, cross-referencing, and source compare are all supported.

If you define the special Programmer Worklist user-defined options, you can run common SVR maintenance options and functions from your worklist panel: see ['3: Check the programmer worklist user-defined options \(*USEROPT\)' on page 16](#) and ['2: Define programmer worklist user-defined options manually' on page 27](#).

The following section describe the initial steps for working with JD Edwards World object types on your worklist. For more information, see "Working with projects and tasks" in the *TURNOVER® Lifecycle Manager User Guide*.

Note TURNOVER® Lifecycle Manager supports Dream Writer, World Writer, and FASTR versions on the worklist when you use type codes JDEDWV, JDEWWV, and JDEFSV. For more information, see ['Managing Dream Writer, World Writer, and FASTR versions' on page 52](#).

Creating a worklist

To start the worklist

Do one of the following:

- Select your TURNOVER® Lifecycle Manager Projects task with option 20 to start PWM and create a worklist for the task.
- On the TURNOVER® Lifecycle Manager main menu, select option 12. The Work with Programmer Worklists panel appears. Press F6 to add a new worklist.
- On a TURNOVER® Lifecycle Manager command line, type STRPWM.

A new worklist is empty and looks like this:

MM/DD/YYYY	11:22:02	Work with Programmer Worklist	Your Company, Inc.
Worklist JDE 0001 JDE Task #1			
Position to Apply filters . . . Y (F17=Filters)			
2=Change 3=Copy 4=Delete 11=Move 12=Compare source 99=Next option			
21=Checkout 22=Change checkout 24=Checkin 25=Task details 26=Conflicts			
Object Chk Nxt			
Object	Attr	Library	Obj Src Out Form FormSts ItemSts Opt
Parameters or command			
====>			
F2=Forms F3=Exit F4=Prompt F5=Refresh F6=Add item F9=Retrieve			
F10=Command entry F11=View2 F19=Inter. 21/46 F23=More options F24=More keys			

Adding items to the worklist

Populate the worklist with items you plan to change or create in order to complete your assigned task. You can add existing objects, or you can add the names of new objects for which you plan to create source code (this is called “reserving” the name).

Adding existing items

Do one of the following:

- Press F20, and then select from the list of JD Edwards World objects in your production partition.
- Press F6 on your new worklist and fill in the appropriate information for an existing object. The panel you see when you do this looks just like the panel for reserving objects, shown in the next topic.

Reserving names for new JD Edwards World objects

New JD Edwards World object names must be reserved for use by a specific programmer. TURNOVER® Lifecycle Manager checks that any names you choose for new objects are both unique and available, and that they match the application's naming mask (if there is one).

- 1 To reserve a name for an object you plan to create, press F6.

The Add Programmer Worklist Item panel appears.

Add Programmer Worklist Item (ADDPWLITM)		
Type choices, press Enter.		
List	> 'JDE 0001'	Name, *SELECT, *ACTIVE
Object	_____	Character value, *FORM
Attribute	_____	*SELECT, *ALL, AFILE, BAS, ...
Checkout item	*NO	*NO, *YES
Description	*OBJD	
Programmer	*LIST	Name, *LIST
Reference	*LIST	*LIST, Char
User defined status	NEW	Character value
Application	*LIST	Character value, *LIST
Release	_____	Number
Version	_____	Number
Create worklist	*NO	*NO, *YES
Form number	*NONE	Number, *NONE
Bottom		
F3=Exit F4=Prompt F5=Refresh F10=Additional parameters F12=Cancel		
F13=How to use this display F24=More keys		

- 2 Type the JD Edwards World object type and attribute. Include a release and version, if applicable.

Add Programmer Worklist Item (ADDPWLITM)		
Type choices, press Enter.		
List	> 'JDE 0001'	Name, *SELECT, *ACTIVE
Object	A0004	Character value, *FORM
Attribute	JDEDD	*SELECT, *ALL, AFILE, BAS, ...
Checkout item	*NO	*NO, *YES
Description	*OBJD	
Programmer	*LIST	Name, *LIST
Reference	*LIST	*LIST, Char
User defined status	NEW	Character value
Application	*LIST	Character value, *LIST
Release	_____	Number
Version	_____	Number
Create worklist	*NO	*NO, *YES
Form number	*NONE	Number, *NONE
Bottom		
F3=Exit F4=Prompt F5=Refresh F10=Additional parameters F12=Cancel		
F13=How to use this display F24=More keys		

- 3 Press Enter.

Specifying function use for IBM i object types

When you add a native object with a JD Edwards World IBM i type code to a worklist, a **Function use** field appears on the Add Programmer Worklist Item panel.

Add Programmer Worklist Item (ADDPWLITM)

Type choices, press Enter.

List	> 'JDE 0001'	Name, *SELECT, *ACTIVE
Object	> A0004	Character value, *FORM
Attribute	> CLPJDE	*SELECT, *ALL, AAUDIT, ABCD...
Checkout item	*NO	*NO, *YES
Description	*OBJD	
Programmer	*LIST	Name, *LIST
Reference	*LIST	*LIST, Char
User defined status	NEW	Character value
Application	*LIST	Character value, *LIST
Release	_____	Number
Version	_____	Number
Create worklist	*NO	*NO, *YES
Form number	*NONE	Number, *NONE
Function use	*SELECT	Character value, *SELECT...
Bottom		
F3=Exit F4=Prompt F5=Refresh F10=Additional parameters F12=Cancel		
F13=How to use this display F24=More keys		

1 In the **Function use** field, type *SELECT.

The Select Function Use panel appears.

Add Programmer Worklist Item (ADDPWLITM)

Type choices, press Enter.

List		
Object		Select Function Use
Attribute		
Checkout item	Member . . .	A0004
Description	Type	RPGJDE
Programmer	1=Select	
Reference		
User defined status	Use	Description
Application	—	Undefined
Release	— 1	Programs
Version	— 110	Video Programs
Create worklist	— 111	File Maintenance
Form number	— 112	Transaction Proc.
Function use	— 113	Inquiry
	— 119	Window Programs
	— 130	Batch Programs
F3=Exit F4=Prompt F5=Refresh More...		
F13=How to use this display F3=Exit F12=Cancel		

Use this panel to define to TURNOVER® Lifecycle Manager the SVR function to use for the JD Edwards World object you are adding to the worklist.

Bypassing the Select Function Use panel

You can bypass the Select Function Use panel by writing a program that defines the SVR function for TURNOVER® Lifecycle Manager to use. Name this program **TURCJFUS**. It must declare these parameters:

PMBR (char, 10)
PTYPE (char, 10)
PUSE (char, 3)
PFLAG (char, 1)

You can find sample source for this program in source file ACLSRC in the TURNOVER® Lifecycle Manager product library.

Put the TURCJFUS program in the library list you will use when adding the JD Edwards World objects to the worklist. TURNOVER® Lifecycle Manager uses the function code that is passed back in the PUSE parameter.

Additional worklist options

MM/DD/YYYY 14:36:14		Work with Programmer Worklist				Your Company, Inc.					
Worklist JDE00001		JDE Task #1									
Position to . . .		Apply filters . . . <u>Y</u> (F17=Filters)									
2=Change 3=Copy 4=Delete 11=Move 12=Compare source 99=Next option											
21=Checkout 22=Change checkout 24=Checkin 25=Task details 26=Conflicts											
		Source		Source		Checkout		Chk Chk			
Object		Attr	Library	File	Programmer	Appl	Rl	Vr	Lv	Xrf	Rel
___ DSPJDE		DSPJDE	PRODTEST	JDESRC		20JD			2		
___ DSPJDE		DSPJDE	ACPTTEST	JDESRC		20JD			1		
___ DSPJDE		DSPJDE	DEVTEST	JDESRC		20JD			1		
___ LFJDE		LFJDE	PRODTEST	JDESRC		20JD			2		
___ LFJDE		LFJDE	ACPTTEST	JDESRC		20JD			1		
___ LFJDE		LFJDE	DEVTEST	JDESRC		20JD			1		
Bottom											
Parameters or command											
====>											
F2=Forms F3=Exit F4=Prompt F5=Refresh F6=Add item F9=Retrieve											
F10=Command entry F11=View3 F19=Inter. 21/46 F23=More options F24=More keys											

To display other standard options, press F23.

You can also create user-defined options that perform JD Edwards World SVR functions against World objects on the worklist. For a list of recommended options, see ['2: Define programmer worklist user-defined options manually' on page 27](#). Press F16 to work with user-defined options for the Programmer Worklist.

TURNOVER® Lifecycle Manager commands for JD Edwards World applications

These TURNOVER® Lifecycle Manager commands provide support for the data-driven type codes.

CHKJDEOBJ: Check JDE Object

This command checks if a JD Edwards World object exists. It returns an error message if the World object does not exist. (This command is similar to the IBM i command, CHKOBJ.)

```

                                Check JDE Object (CHKJDEOBJ)

Type choices, press Enter.

Object . . . . . Character value
Library . . . . . Character value
Object type . . . . . Character value


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

Parameter	Description
Object	For example, the name of a Dream Writer.
Library	The library in which files for the object exist, usually your Common library)
Object type	*JDEDW, *JDEMNU, and so on.

DLTJDEOBJ: Delete JDE Object

This command deletes a JD Edwards World object (deletes the corresponding records from the World files). This command is similar to the IBM i command, `DLTPGM`.

```

                                Delete JDE Object (DLTJDEOBJ)

Type choices, press Enter.

Object . . . . . Character value
Library . . . . . Character value
Object type . . . . . Character value


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

The required parameters are **Object**, **Library**, and **Object type** (see `CHKJDEOBJ` for parameter descriptions).

CPYJDEOBJ: Copy JDE Object

This command copies a JD Edwards World object from one library to another. This command is similar to the IBM i command, CPYF.

Copy JDE Object (CPYJDEOBJ)		
Type choices, press Enter.		
Object		Character value
Object type		Character value
From library		Character value
To library		Character value
Copy option	<u>*REPLACE</u>	*REPLACE, *ADD
Create files	<u>*NO</u>	*NO, *YES
Bottom		
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display		
F24=More keys		

The required parameters are **Object**, **Object type**, **From library**, **To library**, **Copy option** (*ADD, *REPLACE), and **Create files** (*YES, *NO).

This command returns an error message for any of the following conditions:

- The from-object does not exist.
- JD Edwards World files do not exist in to-library, and *NO is specified for **Create files**.
- *ADD is specified for **Copy option**, and the object already exists in the to-library.

CHKOUTJDE: Checkout JDE Object

This command copies the JD Edwards World object to the application's development library and writes a corresponding checkout record in the source checkout control file.

This command is similar to the TURNOVER® Lifecycle Manager CHECKOUT command.

```

Checkout JDE Object (CHKOUTJDE)

Type choices, press Enter.

Object . . . . .
Type code . . . . . _____ JDD, JDDBAS, JDDF, JDDSMP, ...
Application . . . . . _____ CHAR
Level . . . . . _____ Number
Reference . . . . . _____ Char

Additional Parameters

Library . . . . . _____ *APPL *APPL, name
Copy option . . . . . _____ *REPLACE *NONE, *ADD, *REPLACE
Release . . . . . _____ Number
Version . . . . . _____ Number
Programmer . . . . . _____ *USRPRF Name, *USRPRF

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

The required parameters are **Object**, **Type code**, and **Application**.

To see the additional parameters, press F10.

Copy option

You can specify the default value you want to use when you check out an existing JD Edwards World soft-coded type from the worklist.

- 1 Create a data area named TJDECPYOPT (*CHAR 8 and the text "Copy option for checkout of soft coded types") in the TURNOVER® Lifecycle Manager data library (SOFTTURND).
- 2 Populate the data area with *NONE, *ADD, or *REPLACE.
- 3 When you check out from the worklist, TURNOVER® Lifecycle Manager uses the value you have specified in the data area.

Menu language preferences

Defining the language preference

The JD Edwards World menu name can be up to 10 characters, so there is no room to include both the two-character language preference *and* the menu name in the object field in the TURNOVER® Lifecycle Manager database. Therefore, TURNOVER® Lifecycle Manager uses a different type code for each menu language that is supported.

The JDEMNU type code has object type of *JDEMNU. If a type code is defined with an object type that starts with *JDEMNU, TURNOVER® Lifecycle Manager uses positions 8-9 of the object type (blank for type *JDEMNU) to determine the language preference.

For example, to define a type code for managing menus with language preference SP, you might define a type code called JDEMSP with object type of *JDEMNU SP.

Adding an entry file to the TURNOVER® Lifecycle Manager data library

Before defining a type code that has an object type that is not a valid IBM i object type (*PGM, *FILE, and so on), you might have to first add an entry to file TUSRTYPE in the TURNOVER® Lifecycle Manager data library. Display the records in that file to see how it works. TURNOVER® Lifecycle Manager adds some types (*JDD, *JUD, *JVO) to this file automatically, but it provides no method for maintaining that file for other types. Therefore, you have to maintain those other records using DFU or a similar file editor.

Recording SAR numbers in the SVR detail record

Depending on the **Delivery type** information in your CASE profile, TURNOVER® Lifecycle Manager might ask you to enter an SAR number (software action request number) when you edit a JD Edwards World source member.

(If there is no CASE profile entry for your user profile, TURNOVER® Lifecycle Manager checks the **Delivery type** information in the *PUBLIC CASE profile.)

TURNOVER® Lifecycle Manager behaves as follows, according to the information it finds:

Delivery type Action

*NONE	No prompt appears; no SAR number is entered in the SVR detail record.
*LOG	No prompt appears; zeros are entered.
*DFT	No prompt appears; SAR number in CASE profile is entered.
*PROMPT	A window appears so you can enter or change the SAR number. The number you enter is verified against the Work Order System (file F4801) in the SAR File Library specified in the CASE profile.

Setting the JD Edwards World Attention program

Typically, you use the attention feature to return to a JD Edwards World application after you have gone into TURNOVER® Lifecycle Manager using the STRPWM or TURNOVER® Lifecycle Manager command. The default attention program is P98GRP. With this default set, if you press the Attention key to return to the World application, you might find that your original World library is changed when you return: it might include World libraries carried over from the TURNOVER® Lifecycle Manager environment.

To preserve your original JD Edwards World library list, you can set a parameter in your World profile that selects a TURNOVER® Lifecycle Manager program called TP98GRP as the attention program. With TP98GRP set as the attention program, your World library list is restored to its original state when you return to your World application. The TP98GRP program is in the TURNOVER® Lifecycle Manager product library.

7 DBCS versions of JD Edwards software

If you use a DBCS version of JD Edwards software *other* than version 7.3, you might need to recompile some TURNOVER® Lifecycle Manager programs that directly access JD Edwards files, so that these programs can be run over the DBCS version of the JD Edwards files.

If you are running JDE version 7.3, you do *not* need to recompile the programs for DBCS, because TURNOVER® Lifecycle Manager provides recompiled DBCS programs for that version.

TURNOVER® Lifecycle Manager versions dated January 2010 and later

TURNOVER® Lifecycle Manager dated January 2010 and later uses a new method of managing DBCS versions of JDE. If you run a version other than 7.3 you might still need to recompile some programs using the `TCRTJDEDB` command, *but* instead of creating these programs with the same name as existing programs, the command now creates them with different names. This allows you to manage both SBCS and DBCS versions of JDE software using separate application definitions in the same installation of TURNOVER® Lifecycle Manager, without requiring multiple, separate TURNOVER® Lifecycle Manager installations.

Setup

- 1 Follow the directions for setting up alternative environments: see [‘Setting up TURNOVER® Lifecycle Manager for World applications’ on page 13](#).
 - If you are managing only SBCS or only DBCS, create a `*DEFAULT` environment for that version.
 - To manage both SBCS and DBCS, create a `*DEFAULT` environment for the primary version, and then create an alternative environment for the application definition that will manage the secondary version.
- 2 In TURNOVER® Lifecycle Manager, make sure that the libraries containing the DBCS version of the JDE files are *before* any libraries containing SBCS versions in your library list.
- 3 Run the `TCRTJDEDB` command as specified below.

The Target Library for the DBCS programs is the TURNOVER® Lifecycle Manager product library.

Create JDE DoubleByte command (TCRTJDEDB)

The TCRTJDEDB command for compiling the necessary TURNOVER® Lifecycle Manager programs is in the TxxxINST library, where xxx is your TURNOVER® Lifecycle Manager release.

To use this command, you must be in a TURNOVER® Lifecycle Manager environment and have *ALLOBJ authority.

The library containing the JD Edwards DBCS files must be in your library list ahead of any libraries containing JD Edwards SBCS files.

Target library for programs

Specify the library where you want to put the compiled programs.

- Usually, this is the TURNOVER® Lifecycle Manager product library (usually SOFTTURN).
- However, you can put the programs in a separate library that you can save, and then send to other computers that are also using a JD Edwards DBCS environment. If you do this make sure that:
 - The TURNOVER® Lifecycle Manager release date on those computers matches the TURNOVER® Lifecycle Manager release date of the computer on which the programs were compiled
 - The JDE release and cumulative level is the same on each computer.

You must eventually out the compiled programs in the TURNOVER® Lifecycle Manager product library on each computer.

Target release

Specify the target release of the operating system on which you will use the objects you are creating.

Value	Description
*CURRENT	The target release of the operating system currently running on your system.
*PRV	The previous release with modification level 0 of the operating system.
VxRxMx	A release that you specify.

For more information about these values, see the online help for this command.

Replace program

Specify whether you want to create a new program if a program of the same name already exists in the specified library.

Value	Description
*YES	Specify this the first time you run the command. TURNOVER® Lifecycle Manager does not actually replace the object but moves it to an archive library named <i>Thhmmssjjj</i> , where <i>hmmss</i> is the current time and <i>jjj</i> is the last three positions of the job that compiles the programs.
*NO	Specify this for subsequent command runs. This causes TURNOVER® Lifecycle Manager to fail the compile of programs that already exist, and to create only programs that do not exist.

The TCRTJDEDB command compiles several programs. TURNOVER® Lifecycle Manager issues the CRTBNDRPG command for each program, regardless of the success or failure of previous compiles. Also, TURNOVER® Lifecycle Manager moves each existing program to the archive library when you specify *YES, regardless of the success or failure of the compiles. If any of the compiles fail, you can re-run the compile job after fixing the problem that caused the failures (typically library list issues).

Specify your JDE release

Specify the major release number for your JD Edwards software.

The TURNOVER® Lifecycle Manager JD Edwards interface uses different programs, depending on your JD Edwards release. TURNOVER® Lifecycle Manager compiles only programs necessary for your JD Edwards release.

Submit compiles to batch

You can run the compile job in batch or interactively.

Value	Description
*YES	Submit the TCRTJDEDB compile job to batch. TURNOVER® Lifecycle Manager displays SBMJOB command prompts so you can customize the JOBQ, submit time, and so on. Because the library lists are already verified, you cannot change them.
*NO	Run the compiles interactively. Use this value if you have used the TCRTJDEDB command in your own program that is running in batch.



www.softlanding.com

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

www.unicomglobal.com