



The Mainframe Software Partner  
For The Next 50 Years

# Xpediter/CICS

---

## Quick Configuration Guide

**Release 9.4**

Please direct questions about Xpediter/CICS  
or comments on this document to:

**Xpediter/CICS Customer Support**

**<http://go.compuware.com/>**

This document and the product referenced in it are subject to the following legends:

Copyright 1984 - 2015 Compuware Corporation. All rights reserved. Unpublished rights reserved under the Copyright Laws of the United States.

U.S. GOVERNMENT RIGHTS-Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in Compuware Corporation license agreement and as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013(c)(1)(ii) (OCT 1988), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14 (ALT III), as applicable. Compuware Corporation.

This product contains confidential information and trade secrets of Compuware Corporation. Use, disclosure, or reproduction is prohibited without the prior express written permission of Compuware Corporation. Access is limited to authorized users. Use of this product is subject to the terms and conditions of the user's License Agreement with Compuware Corporation.

Xpediter, Xpediter/CICS, Xpediter/*Xchange*, Xpediter/Code Coverage, Abend-AID for CICS, FrontLine, and Compuware are trademarks or registered trademarks of Compuware Corporation.

CICSplex, CICS Transaction Server, DB2, DFSMS, IBM, MQ Series, MVS, MVS/ESA, OS/390, VTAM, WebSphere MQ, and z/OS are trademarks or registered trademarks of International Business Machines Corporation.

Adobe® Reader® is a trademark of Adobe Systems Incorporated in the United States and/or other countries.

All other company and product names are trademarks or registered trademarks of their respective owners.

# Chapter 1.

## Xpediter/CICS Quick Configuration

This *Quick Configuration Guide* is intended for system programmers and database administrators responsible for the configuration of Xpediter/CICS.

This guide provides a checklist to configure and customize Xpediter/CICS once the SMP/E installation has completed (refer to the refer to the *Compuware Installer Mainframe Products SMP/E Installation Guide*). If you need more detailed information, please refer to the online HELP (PF1) and the full *Xpediter/CICS Installation and Configuration Guide*.

---

## Configuration Procedures

The following steps can be used to configure Xpediter/CICS.

**Note:** Before you begin the Xpediter/CICS configuration, it is necessary to verify that Compuware Shared Services (CSS) Release 9.0 or above and License Management System (LMS) Release 3.1 or above have been installed. CSS and LMS are distributed with Enterprise Common Components (ECC), which is included when you order Xpediter/CICS.

### Step 1. Prepare the ISPF Edit Macro (Optional)

XDUPDATE is an ISPF edit macro that can be used to automate the entry of site-specific JCL parameters. Using this macro saves you from having to repeatedly type in the same information and ensures your JCL parameters are always entered correctly.

To use the edit macro, first copy member XDUPDATE from the SMXDSAMP member (as described in the Note above) into your CLIST library. Then enter your site-specific information in the XDUPDATE macro. The macro includes instructions for entering the information. Your CLIST library must be allocated to your ISPF session. You can verify that your CLIST library is accessible to ISPF by entering the command TSO LISTA.

The edit macro can be used during Xpediter installation each time you are instructed to edit JCL. To run it, first open the JCL member for editing. You can then look at the JCL before any changes are made. Type XDUPDATE in the COMMAND field of the ISPF EDIT screen and press Enter.

### Step 2. Integrate Xpediter/CICS with MVS

In this step, which *must* be performed, you will define an Xpediter/CICS APF authorized library to MVS by updating the MVS APF authorized library list. You will also define the Xpediter Service Provider program properties to MVS and define it as an MVS subsystem.

**Notes:**

- The Xpediter/CICS APF authorized library should *not* be included in the operating system's link list. This will allow you to protect the library and ensure only approved users can execute the Xpediter Service Provider.
- If you will be using DBCTL support, do *not* define the Xpediter/CICS APF authorized library as a PDSE, because IMS does not support PDSEs in steplibs.

## Update the MVS APF Authorized Library List

1. For sites using dynamic lists, issue the following MVS operator command to update the MVS APF authorized library list:

**SET PROG=xx** where xx is the suffix of the PROGxx member containing the Xpediter/CICS APF authorized library.

2. For sites using static lists, perform one of the following to update the MVS APF authorized library list:

- Copy the contents of the Xpediter/CICS APF authorized library dataset to an existing MVS APF authorized library. The default dataset name is CPWR.cMXD940.SMXDAUTH where *c* designates the CICS release.

**Note:** The Xpediter/CICS APF authorized library is backward compatible. For more information, see “Step 5. Initialize the Xpediter Service Provider” on page 1-6.

- Use an MVS monitor with APF authorized library update capabilities to dynamically update the MVS APF authorized library list.
- Schedule an IPL. Perform the next step *before* you IPL.

## Define the Xpediter Service Provider Program Properties to MVS

1. Add program DBUGSTC to the MVS program properties table (PPT) member in SYS1.PARMLIB using either of the following methods:
  - Copy SMXDSAMP member SCHEDXD to SYS1.PARMLIB, then add XD to the SCH=(xx) directive of IEASYSxx in SYS1.PARMLIB. For example, change SCH=00 to SCH=(00,XD).
  - Append SMXDSAMP member SCHEDXD to your SCHEDxx member in SYS1.PARMLIB.
2. Update the MVS PPT by issuing one of the following MVS operator commands:

**SET SCH=(xx,XD)** where xx is the suffix of the SCHEDxx member used during the last IPL and XD is the suffix of the SCHEDxx member which contains the MVS PPT entry for DBUGSTC.

**SET SCH=xx** where xx is the suffix of the SCHEDxx member which contains the MVS PPT entry for DBUGSTC.

## Define the Xpediter Service Provider as an MVS Subsystem

You may optionally specify the Xpediter/CICS subsystem ID in SYS1.PARMLIB(IEFSSNxx) for system documentation purposes using either of the following methods:

- Copy SMXDSAMP member IEFSSNXD to SYS1.PARMLIB, then add XD to the SSN=(xx) directive of IEASYSxx in SYS1.PARMLIB. For example, change SSN=00 to SSN=(00,XD).
- Append SMXDSAMP member IEFSSNXD to your IEFSSNxx member in SYS1.PARMLIB.

You are not required to perform an IPL to use the Xpediter Service Provider. If an MVS subsystem entry has not already been defined, Xpediter/CICS will build one dynamically during initialization.

### Notes:

- Make sure the MVS Performance Group used by the Xpediter Service Provider is above CICS and below VTAM to prevent runtime abends.

- For additional information, refer to Chapter 7, “Xpediter Service Provider” in the *Xpediter/CICS Installation and Configuration Guide*.

### Step 3. Update the CICS Resource Definitions

This step explains how to update your site’s CICS resource definitions.

Observe the following **general** considerations:

- Some Xpediter/CICS control blocks are now GETMAINed in accordance with the TASKDATALOC specified for the transaction being tested. Xpediter’s use of below the line storage can be reduced by specifying TASKDATALOC=ANY for your eligible transactions.
- Xpediter/CICS does not support programs with RELOAD:YES in the program resource definition. Remove this option for programs that you plan to debug with Xpediter.
- Do not define temporary storage queues with an XP prefix as remote.
- In DFHSITxx, do not specify SRT=NO.
- Unless the documentation packaging utility has been disabled by changing the value of global parameter RECREATE to NONE, either in DBUGGBL or as a global table override, specify SPOOL=YES as a SIT parameter.
- If your site will be initializing Xpediter/CICS using PLT startup, observe the following conditions:
  - Add the Xpediter/CICS program DBUGPPM2 to the PLT program initialization (PLTPI) table.
  - Shutdown of Xpediter/CICS via the PLT is supported. To enable this support, add program DBUGPLTS to the PLTSD table. In your CICS shutdown PLT, *before* the line reading DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM, add the following entry:

```
DFHPLT TYPE=ENTRY ,PROGRAM=DBUGPLTS
```

After adding the entry, reassemble and relink your PLTSD module. Do **not** put DBUGPPM2 in the PLTSD table.

#### Notes:

- Modification of the PLTSD *will* be required if the Xpediter Script Facility or File Utility Audit Trail is enabled later in this installation procedure.
- When Xpediter/CICS starts the DBPA transaction during PLT post-initialization, the userID assigned is the same as the userID assigned to the CPLT transaction.
- Compuware highly recommends the use of RENTPGM=PROTECT.

### Using RDO to Update Resource Definitions

1. Submit the supplied JCL member DBCRDO to batch the required CEDA transactions. Refer to Table 1-1 and use the SYSIN in resource members CSDXDFIL and CSDXDTRN. Also include CSDXDPRG if autoinstall will not be used to install Xpediter programs at your site.

#### Notes:

- If your site is not using Xpediter/Code Coverage, edit the CSDXDFIL member and comment out the definitions for XVTCEXTR (the Code Coverage extract dataset), XVTCEXTJ (the Code Coverage JCL template dataset), and XVTCSPEC (the Code Coverage Specifications Archive dataset).
- If your site is using BAS, you may use the furnished JCL member CSD2BAS to create an input file for the BATCHREP view in the CICSplex SM End User Interface.

- LSRPOOLNUM or LSRPOOLID may be specified as appropriate. Be aware that for file control update requests, CICS does not support the use of LSRPOOLNUM(NONE) or LSRPOOLID(NONE) while Transaction Isolation is active. Beginning with CICS TS 4.1 this restriction is enforced with an AFDK abend.
- 2. If your site is using RDO to define DB2 resources, use the SYSIN in resource member CSDXDDB2 to batch the CEDA transactions that define the entries required for the DB2 File Utility. Refer to “Step 19. Implement Support for the DB2 File Utility (Optional)” in Chapter 2 of the *Xpediter/CICS Installation and Configuration Guide*.
- 3. If your site is using Xpediter’s CICSplex TOR support, use the SYSIN in resource member CSDXDDYN to batch the CEDA transactions that define the entries required. Refer to “Installation in a CPSM Dynamic Routing Environment” in Chapter 10 of the *Xpediter/CICS Installation and Configuration Guide*.
- 4. The Xpediter/CICS resource group created in this step must be added to a CICS group list, which in turn is specified in the system initialization parameter GRPLIST.
- 5. CSDXDFIL contains an ENQMODEL new in Release 8.1. Ensure that this ENQMODEL definition is added to all CICS regions that share the same Xpediter/CICS Profile Dataset.

**Note:** If you change any of the Xpediter transaction IDs before submitting the JCL, they must also be changed in the Xpediter/CICS global table.

**Table 1-1.** Resource Members

<b>SYSIN Member for JCL Member DBCRDO or CSD2BAS</b>
CSDXDFIL
CSDXDPRG (omit if using autoinstall for Xpediter programs)
CSDXDTRN
CSDXDDB2 (DB2ENTRY and DB2TRAN entries for DB2 sites only)
CSDXDDYN (if using Xpediter’s CICSplex TOR support)

### ***Program Autoinstall***

Xpediter/CICS fully supports autoinstall for user programs, mapsets, and partitionsets.

**If your site has elected to autoinstall Xpediter programs, you must make the following changes to the installation process:**

- Be sure to omit SYSIN member CSDXDPRG. Member CSDXDTRN contains a single program definition for program DBUPGADX, to install the Xpediter/CICS Program Autoinstall Exit program. All other Xpediter/CICS programs are autoinstalled by this exit.
- DBUPGADX is the Xpediter/CICS autoinstall exit for programs. DBUPGADX is designed to work in conjunction with any current Program Autoinstall Exits, so your Autoinstall Exit programs do not need to be changed.
- To implement the Xpediter/CICS Program Autoinstall Exit, alter the system initialization table (SIT) parameters PGAIEXIT and INITPARM as follows by either modifying the SIT or by using SIT overrides:
  - **PGAIEXIT**  
To ensure that the Xpediter/CICS exit gains control before your normal Program Autoinstall Exit, change or override this parameter to PGAIEXIT=DBUPGADX. If you are currently using program autoinstall, take your current exit name and substitute it for DFHPGADX in the INITPARM parameter added below.
  - **INITPARM**

Add DBUPGADX='DFHPGADX' to your existing INITPARM parameters. For example, you would change or override INITPARM=(*current string*) to INITPARM=(DBUPGADX='DFHPGADX',*current string*). If you do not currently use INITPARM, code INITPARM=(DBUPGADX='DFHPGADX').

DFHPGADX is the Program Autoinstall Exit that will be linked to if no program name matches are found in DBUPGADX. Change the string DFHPGADX to the name of the exit that you want to have control.

As mentioned above, if you are already using program autoinstall, you should replace DFHPGADX with the string previously used on your PGAEXIT parameter. If your old SIT parameters were PGAEXIT=OLDPAEXT and INITPARM=(*current string*), you would change or override them to PGAEXIT=DBUPGADX and INITPARM=(DBUPGADX='OLDPAEXT',*current string*).

You may also change this added parameter to DBUPGADX='LAST'. This would indicate to DBUPGADX that no other exits are called if the program being autoinstalled is not an Xpediter/CICS program.

#### Notes:

- If you omit or miscode the INITPARM, the default action is to link to DFHPGADX.
- If PGAEXIT=DBUPGADX and INITPARM=(DBUPGADX='OLDPAEXT'), where OLDPAEXT is written in a high-level language, IGZINSH (an IBM-supplied program) needs to be pre-installed in the CICS region.
- If Xpediter/CICS is not initialized in the CICS region when a user tries to start an Xpediter session, Xpediter will invoke the initialization process. If the user is using the Xpediter/Eclipse interface, the program definitions for DBUGSOCK and XDPIMIRS must be already installed in the CICS region. If Xpediter/Eclipse is used at your site, uncomment the DBUGSOCK and XDPIMIRS definitions supplied in the SMXDSAMP member CSDXDTRN.

## Setting Exit Program Priorities

If you have more than one XPCTA exit program that sets PCUE\_BRANCH\_ADDRESS (the resume address parameter), you must decide on the order of priority. The last exit program enabled is the last to get control and is also the last to move the address into the resume address field. The program only branches to the last resume address. You must ensure that all exit programs correctly check and update the return code fields. The XPCTA sample (DFH\$PCTA) must be modified to be used with multiple XPCTA exit programs, because it does not check for return codes from any previous exits.

**Note:** In order to trap CICS abends, Xpediter/CICS must be the *last* XPCTA exit program to change the resume address.

If you have more than one XPCFTCH exit program that sets PCUE\_BRANCH\_ADDRESS (the resume address parameter), you must decide on the order of priority. The last exit program enabled is the last to get control and is also the last to move the address into the resume address field. The program only branches to the last resume address. You must ensure that all exit programs correctly check and update the return code fields.

**Note:** In order to trace, monitor, or code cover CICS transactions, Xpediter/CICS must be the *last* XPCFTCH exit program to change the resume address unless the last exit is for Xchange/CICS.

## Step 4. Creating and Managing the Xpediter TP Configuration File (TPCONFIG)

The Xpediter TP configuration file is required to provide various types of information to the mainframe side of the Xpediter/CICS and Xpediter/Eclipse products. The Xpediter

Service Provider requires this dataset to identify all of the other Service Providers, in other LPARs, that are to share CICSplex information. Xpediter/CICS also requires this dataset to identify all the CICS regions that should be contacted by the Batch Interface to CICS NEWCOPY. For usage with Xpediter/Eclipse, see the documentation in the *Compuware Workbench Installation Guide*.

**Note:** If you have been using the TPCONFIG file, you may continue to use your existing dataset.

If your configuration file is on shared DASD that can be accessed by all LPARS, then only one dataset, with one member, is required. The structure of entries within a member is such that one member may provide all of the necessary information for Xpediter/Eclipse, all Xpediter Service Providers, and the Xpediter/CICS Batch Interface to CICS NEWCOPY. However, particularly for the Batch Interface to CICS NEWCOPY, you may find it easier to have multiple members in one dataset or even multiple datasets.

The configuration file should have its access security set so that all users can read it, but can be updated only by those to whom you have granted edit access.

The configuration file dataset does not have to be in PDSE format, however a PDSE is recommended to avoid compression issues if you frequently modify the member(s).

An example of a configuration file member is provided in the Xpediter/CICS sample dataset (member name MBRTPCNF). The following is a summary of the steps needed to create and use the configuration file:

1. Edit and submit the sample JCL member JCLTPCNF to allocate the configuration file and copy the sample configuration member.
2. Customize the configuration member to your installation needs.

This dataset will be referred to as the TPCONFIG dataset in “Step 5. Initialize the Xpediter Service Provider” on page 1-6 and in “Step 16. Install and Customize the Batch Interface to Xpediter/CICS NEWCOPY (Optional)” of Chapter 2 in the *Xpediter/CICS Installation and Configuration Guide*.

## Step 5. Initialize the Xpediter Service Provider

Before proceeding, make sure you have completed the procedures in “Step 2. Integrate Xpediter/CICS with MVS” on page 1-1.

As shown in Table 1-2, this release of Xpediter can only be used with this version of the Xpediter Service Provider. For those sites using multiple releases of Xpediter/CICS, this release of the Service Provider is backward compatible with Releases 9.3, 9.2, 9.1, and 9.0 of Xpediter/CICS.

**Table 1-2.** Xpediter Service Provider Compatibility

Xpediter Release	Service Provider Version				
	9.0	9.1	9.2	9.3	9.4
9.0	•	•	•	•	•
9.1		•	•	•	•
9.2			•	•	•
9.3				•	•
9.4					•

1. Edit and enable SMXDSAMP member XDSSPROC as follows:



- a. Make sure the Xpediter/CICS APF authorized dataset name in the STEPLIB DD statement matches the name used at your site. The default is SMXDAUTH member.
- b. Make sure the SSID parameter value, which specifies the four-character subsystem identifier for the Xpediter Service Provider subsystem, matches the value of global parameter SUBSYS used at your site. The default is XDSS.
- c. If you do not plan to share CICSplex information across LPARs, then delete the TPCONFIG DD statement and related parameters MEMBER and PORT from XDSSPROC, delete parameters MEMBER and PORT from XDSSJCL, and skip the rest of this substep.

The Xpediter Service Provider uses TCP to communicate to other Service Providers. If you want this instance of the Service Provider to share CICSplex information with other Service Providers across LPARs, then customize the dataset name in the TPCONFIG DD statement and the two related parameters, MEMBER and PORT. Specify your TPCONFIG member name (from “Step 4. Creating and Managing the Xpediter TP Configuration File (TPCONFIG)” on page 1-5) in the XDSSPROC MEMBER parameter. Your TCP Systems Programmer can provide you with a Port Number and either a Domain Name or an IP Address for each instance of the Service Provider that you create. Specify the Port Number for this instance of the Service Provider in the PORT parameter. Using the information provided by your TCP Systems Programmer, add an entry to the TPCONFIG member. See SMXDSAMP member MBRTPCNF for an example XDSS entry.

- d. Copy member XDSSPROC from SMXDSAMP member to a JCL procedure dataset defined to JES.
2. The Xpediter Service Provider subsystem can be executed as a batch job or as a started task.
    - To run the subsystem as a batch job:
      1. Edit member XDSSJCL provided in SMXDSAMP member as required. Make sure the SSID parameter value, which specifies the four-character subsystem identifier for the Xpediter Service Provider, matches the name used at your site. The default is XDSS.
      2. Submit XDSSJCL.
    - To run the subsystem as a started task, issue the following operator command:

**S XDSSPROC{,SSID=xxxx}** where {,SSID=xxxx} is an optional parameter used to override the default subsystem identifier, XDSS, with the identifier used at your site.

If you would like the Xpediter Service Provider to be initialized automatically during MVS startup, add the following start command to SYS1.PARMLIB member COMMNDxx:

**S XDSSPROC,SSID=xxxx** where xxxx is the subsystem identifier. XDSS is the default.

**Note:** To use the new TCP link between Service Providers, the start command for the Xpediter Service Provider must come **after** your start of TCPIP.

For more information on the subsystem, see Chapter 7, “Xpediter Service Provider” in the *Xpediter/CICS Installation and Configuration Guide*.

For information on subsystem security, refer to “Step 6. External Security Considerations” on page 1-8.

**Note:** The MVS Performance Group for the Xpediter Service Provider must be higher than CICS and less than VTAM to prevent runtime abends.

## Step 6. External Security Considerations

To help you to set up security groups for external security, the following tables list the Xpediter/CICS and Xpediter/Code Coverage transactions and their uses. The tables also identify started transactions and provide recommended external security settings.

The terms used in the “Security Recommended” column are defined as follows:

- **NO** - All users of Xpediter, including the userID associated with PLTPI programs, should be permitted to execute these transactions.
- **YES** - These transactions provide capabilities beyond the needs of a typical end user of Xpediter. Access may be limited to those responsible for installation, configuration, etc.
- **EITHER** - No recommendation.

**Table 1-3.** Xpediter/CICS Transactions

Trans	Program	Use	Started	Security Recommended
DBFL	DEBUGFILE	File utility	•	NO
DBPA	DEBUGPPM2	PLT startup	•	NO
DBXG	DEBUGSP00	Remote traps	•	NO
DMAP	DEBUGMAPS	Display BMS maps		EITHER
NEWC	DEBUGNEWC	Newcopy (PHASEIN) programs		EITHER
XACH	CWDEMACCH	Demo program		EITHER
XASL	CWCEEASM	Demo program		EITHER
XASM	CWDEMASM	Demo program		EITHER
XCB2	CWDEMCB2	Demo program		EITHER
XCBN	CWDEMCBN	Demo program		EITHER
XCCC	CWDEMC	Demo program		EITHER
XCCH	CWDEMCCCH	Demo program		EITHER
XDBP	DEBUGDBPA	Process DBPA input		YES
XIVP	DEBUGINST	Installation Verification Program		EITHER
XLGI	DEBUGLOGC	File Utility logging	•	NO
XLOG	DEBUGLOGM	Activate, terminate, or switch datasets		YES
XPCH	CWCEMPCH	Demo program		EITHER
XPCI	DEBUGCTRL	Remote Operations Command Interface (ROCI)	•	EITHER
XPED	DEBUGSP00	Access Xpediter		EITHER
XPFS	DEBUGCSFS	SLS Dataset Services	•	NO
XPGD	XDPIMIRS	Secure Xpediter/CICS access from Xpediter/Eclipse		EITHER
XPLE	CWDEMPE	Demo program		EITHER
XPLI	CWDEMPL	Demo program		EITHER
XPNO	DEBUGEND0	Automatic session termination	•	NO
XPNC	DEBUGCTRL	Newcopy (PHASEIN) program from batch		EITHER
XPND	DEBUGEND	Automatic session termination	•	NO
XPOF	DEBUGDBXX	Shutdown Xpediter		EITHER
XPRT	DEBUGSP00	Access Xpediter		EITHER

**Table 1-3.** Xpediter/CICS Transactions

Trans	Program	Use	Started	Security Recommended
XPSP	DEBUGSP00	Access Xpediter with system facilities		YES
XPWD	DEBUGWRAM	Disable Xpediter's dynamic transaction routing exit		YES
XPWI	DEBUGWRAM	Enable Xpediter's dynamic transaction routing exit		YES
XREL	DEBUGNEWC	Internal Xpediter transaction		NO
XROI	DEBUGROIC	Interface to License Management System	•	NO
XSIT	DEBUGSIT	Process global overrides		YES
XSOC	DEBUGSOCK	Xpediter/Eclipse connections		NO
XSTA	DEBUGSTA0	Access Xpediter in multiple AORs		EITHER
XZCH	CWDEMZCH	Demo program		EITHER

**Table 1-4.** Xpediter/Code Coverage Transactions

Trans	Program	Use	Started	Security Recommended
XVCC	XVTCMGR	Code Coverage definition update		YES
XVKP	XVTCEXTR	Internal Code Coverage transaction	•	NO
XVSC	XVTCMGR	Code Coverage definition update		YES
XVSI	XVTCSIT	Process global overrides		YES
XVTQ	XVTCCIN	Internal Xpediter/Code Coverage quiesce tran ID	•	NO

**CAUTION:**

If external security checking is enabled (SEC=YES and XPCT=YES in the CICS System Initialization Table) for the started transactions identified in Table 1-3 and Table 1-4, CICS calls the external security manager using the default CICS resource class name CICSPT prefixed with A or B. This is done to verify that the userID associated with the transaction is authorized to use started transactions. Please refer to the CICS *System Definition Guide* and your external security manual for more information.

Incorrect security settings for the XPFS transaction can cause a NO SOURCE AVAILABLE condition for all program listings accessed via Shared Directory or DDIO datasets and can also cause the SLS Datasets (9.L) screen to function improperly.

Incorrect security settings for the XREL transaction can cause erroneous failure of the new copy function for programs where breakpoints have been removed.

Incorrect security settings for the XVKP transaction can cause failure of periodic extract of data for Xpediter/Code Coverage.

The following table lists the results of assigning user security to Xpediter/CICS started transactions.

**Table 1-5.** User Security for Xpediter/CICS Started Transactions

Trans	Program	Use	UserID
DBFL	DEBUGFILE	File utility	Assigned the userID of the terminal accessing the file utility.
DBPA	DEBUGPPM2	PLT startup	Assigned the same userID as the CPLT transaction.
DBXG	DEBUGSP00	Remote traps	Assigned the userID of the trapping terminal.
XLGI	DEBUGLOGC	Activate, terminate, or switch logging	Assigned the userID of the terminal accessing transaction XLOG.

**Table 1-5.** User Security for Xpediter/CICS Started Transactions

Trans	Program	Use	UserID
XPCI	DEBUGCTRL	Remote Operations Command Interface (ROCI)	Refer to "Step 27. Install and Customize the Remote Operations Command Interface (Optional)" in Chapter 2 of the <i>Xpediter/CICS Installation and Configuration Guide</i> for security considerations.
XPFS	DEBUGCSFS	SLS Dataset Services	Assigned the userID of the Xpediter/CICS initiator or DFLTUSER. If initiated via PLTPI, assigned PLTPIUSR or the CICS region userID.
XPGD	XDPIMIRS	Xpediter/Eclipse	Refer to "Step 13. Install and Customize for Xpediter/Eclipse (Optional)" on page 1-17.
XPNO	DEBUGEND0	Automatic session termination	Assigned the userID of the terminal at which it runs (when started via autoinstall terminal delete).
XPND	DEBUGEND	Automatic session termination	Same as XPNO.
XREL	DEBUGNEWC	Release held programs	Runs when all breakpoints are removed from a program. Assigned the userID of the terminal for which the breakpoints were removed.
XROI	DEBUGROIC	Interface to License Management System	Assigned the userID of the Xpediter/CICS initiator. If initiated via PLTPI, assigned PLTPIUSR or DFLTUSER userID. If initiated via terminal, assigned userID of the terminal.

**Notes:**

- If automatic session termination is performed via the XSNOFF global exit, XPNO and XPND must not be secured.
- When Xpediter/CICS starts the DBFL and XREL transactions internally, the current userID will be propagated to either the DBFL or XREL transaction.

The following table lists the results of assigning user security to Xpediter/Code Coverage transactions.

**Table 1-6.** User Security for Xpediter/Code Coverage Started Transactions

Trans	Program	Use	UserID
XVKP	XVTCEXTR	Code Coverage auto-extract transaction	Assigned the userID of the Xpediter/CICS initiator or DFLTUSER. If initiated via PLTPI, assigned PLTPIUSR or the CICS region userID.
XVTQ	XVTCCIN	Code Coverage quiesce transaction	Assigned the userID of the Xpediter/CICS initiator or DFLTUSER. If initiated via PLTPI, assigned PLTPIUSR or the CICS region userID.

## Xpediter Service Provider Security

The delivery of the default Subsystem Security Exit (DEBUGSSX) has changed, beginning with Release 8.2. The default exit is now delivered with the load module name DEBUGSSXX in SMXDAUTH. Therefore, a Subsystem Security Exit will be used only if you take action. If you need to use a Subsystem Security Exit, you can rename DEBUGSSXX to DEBUGSSX or create your own customized version. An example SMP/E USERMOD to rename the module is in member SSSXUMOD in SMXDSAMP.

If you are using an external security manager, such as RACF, at your site, you may need to customize the Subsystem Security Exit (DEBUGSSX) and/or define access rules to allow a CICS address space to access the Xpediter Service Provider Subsystem. Use the sample JCL provided in SMXDSAMP member DBCSSX (Figure 1-1 on page 1-11) to assemble and link-edit DEBUGSSX.

Figure 1-1. Sample DBUGSSX Assembly JCL Member (DBCSSX)

```

***** Top of Data *****
//*====>INSERT JOBCARD HERE <===JOBCARD
//*****
//ASM      EXEC PGM=ASMA90,
//          PARM=('LIST,OBJECT,TERM,LINECOUNT(58)')
//SYSTEM  DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSLIB  DD  DISP=SHR,DSN=SYS1.MACLIB
//          DD  DISP=SHR,DSN=SYS1.MODGEN
//SYSLIN  DD  DISP=(NEW,PASS),DSN=&&OBJLIB,
//          UNIT=SYSDA,SPACE=(CYL,(3,1))
//SYSUT1  DD  UNIT=SYSDA,SPACE=(CYL,(3,1)),DISP=(,DELETE)
//SYSUT2  DD  UNIT=SYSDA,SPACE=(CYL,(3,1)),DISP=(,DELETE)
//SYSUT3  DD  UNIT=SYSDA,SPACE=(CYL,(3,1)),DISP=(,DELETE)
//SYSPUNCH DD DUMMY,DCB=BLKSIZE=80
//SYSIN   DD  DISP=SHR,HERES SOME MORE
//          DSN=CPWR.cMXD940.SMXDSAMP(DBUGSSX)    <-- CHECK DSN
//*****
//LINKEDIT EXEC PGM=IEWL,
//          PARM='LIST,XREF,MAP,RENT',
//          COND=((5,LT,ASM))
//SYSPRINT DD  SYSOUT=*
//SYSUT1  DD  UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSLIN  DD  DISP=(OLD,DELETE),DSN=&&OBJLIB,
//          UNIT=SYSDA,VOL=REF=*.ASM.SYSLIN
//          DD  *
ENTRY DBUGSSX
NAME DBUGSSX(R)
//SYSLMOD DD  DISP=SHR,
//          DSN=CPWR.cMXD940.SMXDAUTH            <-- CHECK DSN
//

```

During Xpediter Service Provider Subsystem initialization, the optional DBUGSSX exit is fetched from the Xpediter/CICS APF authorized library. DBUGSSX is called during address space connect processing to determine whether the requester is authorized to connect to the Xpediter Service Provider Address Space. **If DBUGSSX is not available, all CICS address space connections will be granted.**

Without any customization, DBUGSSX performs a RACROUTE authorization check for READ access against a model profile dataset resource class. The model profile dataset is named XPEDITER.SUBSYS.*ssid.cccccc* where *ssid* is the Xpediter Service Provider Subsystem Identifier and *ccccc* is the requester address space job or started task name.

### Resource Rules

Generic resource rules apply to the DBUGSSX-generated resource names. For the most complete protection, a site can define specific resource rules for each resource name that may be generated, then only grant the desired access to those resources to selected userIDs and groups (address spaces).

Another alternative is for a site to define a resource rule that denies all access by default and is always applied unless some other, more restrictive, rule is applied. The rule would be specified as XPEDITER.SUBSYS.\*\* and defined as having no universal access (UACC=NONE). No userIDs or groups would have READ access to it.

More specific resource rules could then be defined, allowing access on an individual basis. In one possible scenario, a site that has CICS production processing regions with job names of PRODxxxx and test processing regions with job names of TESTxxxx might want to protect its production regions from use of the Xpediter Service Provider Subsystem, while retaining the capability of using it in their test regions. To accomplish this, first the site would define XPEDITER.SUBSYS.\*\* with UACC=NONE and not grant

READ access to any userIDs or groups. Then XPEDITER.SUBSYS.\*.TEST\* would be defined with UACC=NONE, and only selected userIDs and groups (address spaces) would be given the necessary READ access.

With these rules, site security retains complete control over which address spaces are able to use the Xpediter Service Provider Subsystem.

## Step 7. Update the CICS Startup JCL

Change your site's CICS startup JCL as follows:

1. Make the CSS load library accessible to Xpediter/CICS by adding it to the DFHRPL concatenation in the CICS startup JCL. The CSS load library is named CPWR.LCX $nnn$ .SLCXLOAD ( $nnn$  is the release of CSS) and is included in the installation of Enterprise Common Components.

To enable Xpediter's CICS Release Support, add the appropriate load library listed in Table 1-7 to the DFHRPL concatenation.

**Note:** Dynamic LIBRARY resource definitions may be used instead of the static DFHRPL concatenation. For more information, see "Support for CICS Dynamic LIBRARY Capability" in Chapter 1 of the *Xpediter/CICS Installation and Configuration Guide*.

**Table 1-7.** Load Libraries for CICS Release Support

CICS Release	Dataset Name
CICS TS 3.1	CPWR.MMXD940.SMXDO64L
CICS TS 3.2	CPWR.NMXD940.SMXDO65L
CICS TS 4.1	CPWR.OMXD940.SMXDO66L
CICS TS 4.2	CPWR.PMXD940.SMXDO67L
CICS TS 5.1	CPWR.QMXD940.SMXDO68L
CICS TS 5.2	CPWR.RMXD940.SMXDO69L

2. SLS (source listing support) datasets are associated with ddnames derived from the values of global parameter SLSDDN. These parameter values are composed of a 4-character prefix and a 3-digit suffix. The default is SLSF001. The maximum number of ddnames is specified by the global parameter NBRSLs. For more information, see Chapter 4, "Global Parameters Table" in the *Xpediter/CICS Installation and Configuration Guide*. Allocate one or more SLS datasets to CICS using one or any combination of the following three methods:
  - Add a DD statement and/or RDO definition for each SLS standard DDIO dataset or shared directory to the CICS startup JCL. Shared databases are dynamically allocated as required.
  - Add an RDO file definition with a DSNAME parameter for each SLS standard DDIO dataset or shared directory to the CICS startup JCL. Shared databases are dynamically allocated as required.
  - Enable dynamic allocation by using global parameter SLSDDYN and naming the SLS datasets as required for that feature. See "SLSDDYN" in Chapter 4 of the *Xpediter/CICS Installation and Configuration Guide*.

**Notes:**

- If you make a change to your SLS file that requires closing then opening it while CICS is active, you must make that change from the SLS Datasets screen (9.L).
- If you use a non-IBM dynamic allocation product, ensure that the source listing datasets are excluded from automatic opens performed by these products when

the dataset is allocated. If you fail to do so, you may be unable to access source listings through Xpediter/CICS.

- If you are installing Xpediter/CICS for the first time, the optional datasets described in the remainder of this step have not yet been created.
3. If you are not dynamically allocating CICS datasets and plan on using the profile file to be created in “Step 8. Allocate the New Profile File (DBCDEFPP) (Optional)” on page 1-13, add a DD statement for it. The default name is DBUGPRF. Although the profile file is optional, Compuware highly recommends its use.
  4. Add a DD statement for the global table parameter override dataset to be created in “Step 10. Create the Global Table Parameters Override Dataset (Optional)” on page 1-16. The default ddname is XDGBLINP. Refer to Chapter 5, “Overriding Global Parameters” in the *Xpediter/CICS Installation and Configuration Guide* for complete details.
  5. Unless the documentation packaging utility has been disabled by changing the value of global parameter RECREATE to NONE, either in DBUGGBL or as a global table override, you must do one of the following:
    - Add a DD statement for the Recreate JCL member DBUGUD to be allocated in “Step 11. Allocate and Initialize the Recreate Dataset (DBUGUD) (Optional)” on page 1-16.
    - Use the DSNNAME parameter of the CSD resource definition for the Recreate JCL member DBUGUD.

## Step 8. Allocate the New Profile File (DBCDEFPP) (Optional)

Release 8.1 introduced a new format profile file that enables users to save and restore trap settings. This member contains a utility that creates a new format Profile file and optionally converts an existing old format Profile file.

Installing the profile file enables you to individually define PF key settings, footing options, scroll values, and trap and trace options.

### Notes:

- If you reformatted the profile dataset with Release 8.1, you may continue to use your existing dataset.
- Although this keyed sequential dataset is optional, Compuware recommends its use.
- If you have been using the profile file, you may continue to use your existing dataset. However, the ability to save and restore traps and also associate a profile with a userID only exists with the new format dataset.
- An ENQMODEL for the profile dataset, included in sample member CSDXDFIL, was new to Release 8.1. (It was included in PTFs for previous releases.) Ensure that this ENQMODEL definition is added to all CICS regions that share the same Xpediter/CICS Profile Dataset. This ensures that ENQs done for the Profile File are recognized across all CICS regions in the sysplex.
- Also see “PROFUSR” in Chapter 4 of the *Xpediter/CICS Installation and Configuration Guide*. If you are currently using PROFUSR=NO, you may want to reconsider this setting.

Define a VSAM KSDS file with the attributes shown for SMXDSAMP member DBCDEFPP in Figure 1-2 and Figure 1-3. This member contains the JCL for the profile options.

The resource definition for DBUGPRF is installed as part of “Step 3. Update the CICS Resource Definitions” on page 1-3.

Figure 1-2. Defining the Profile File (DBCDEFPPF) - Part 1 of 2

```

/* DBCDEFPPF - JCL TO ALLOCATE/CONVERT AN XPEDITER/CICS PROFILE FILE.
/*
/* ---
/* If you already have a new-format profile file, then this entire
/* job can be skipped.  An old-format file has a keylength of 8 in
/* the beginning of the record. A new-format file has a keylength
/* of 21 later in the record.
/* ---
/* All Xpediter/CICS releases since 7.6 can use both the old-format
/* and the new-format of profile file.
/* ---
/* The following JCL will allocate the Xpediter/CICS profile file,
/* then convert an old-format profile file to the new format,
/* then optionally "swap" names of your old and new profile files.
/* ---
/* Supply the VOLSER, DSN, and SPACE parameters.
/* ---
/* Read the comments before each JOBSTEP for further
/* customization requirements.
/* ---
/*
/*
/* -----
/* The AMS step is required.
/* ---
/* Change parameters to conform to your installation standards.
/* -----
/*
/*AMS      EXEC PGM=IDCAMS
/*SYSPRINT DD SYSOUT=*
/*SYSIN    DD *

/* Define new profile dataset. Do not reuse production DSN */

DEFINE CLUSTER
(NAME(XPED.NEW.PROFILE.FILE)          /* SUPPLY DSN      */ -
 VOLUMES(?????)                      /* SUPPLY VOLSER  */ -
 CYLINDERS(2 2)                       /* SUPPLY SPACE   */ -
 FREESPACE(50 25)                    /* DO NOT MODIFY */ -
 RECORDSIZE(4081 4081)               /* DO NOT MODIFY */ -
 SHAREOPTIONS(4 3)                   /* DO NOT MODIFY */ -
 UNIQUE                              /* DO NOT MODIFY */ -
 KEYS(21 1314))                      /* DO NOT MODIFY */ -
DATA( NAME(XPED.NEW.PROFILE.FILE.DATA))
INDEX(NAME(XPED.NEW.PROFILE.FILE.INDEX))
/*
/*

```



Figure 1-3. Defining the Profile File (DBCDEFPPF) - Part 2 of 2

```

/* -----
/* The CONVERT step is required only if you have an old-format
/* profile file to convert. If you do not have an old-format profile
/* file to convert, then delete this step and the RENAME step (only
/* the AMS step should be used).
/* ---
/* If you do have an old-format profile file to convert, then ...
/* (1) Change the STEPLIB to the name of your Xpediter/CICS loadlib
/*     (any CICS release).
/* (2) Provide the DSName of your old-format profile file in
/*     the "DSN=.." field of the PROFIN statement.
/* (3) Change the PROFOUT DSName to the name used in the AMS step.
/* -----
/*
/*CONVERT EXEC PGM=DBUGCNV1
/*STEPLIB DD DISP=SHR,DSN=CPWR.cMXDnnn.SMXD0ccL /* CHECK DSN */
/*PROFIN DD DISP=SHR,DSN=XPED.PROD.PROFILE.FILE
/*PROFOUT DD DISP=SHR,DSN=XPED.NEW.PROFILE.FILE
/*
/*
/* -----
/* The RENAME step is optional.
/* ---
/* If you had an old-format profile file, and all CICS regions
/* that reference it are shutdown now, then you may RENAME the
/* old-format file and RENAME the new-format file to that old
/* name.
/* -----
/*
/*RENAME EXEC PGM=IDCAMS,COND=(0,NE,CONVERT)
/*SYSPRINT DD SYSOUT=*
/*SYSIN DD *

/* Rename current production dataset */

ALTER                -
  XPED.PROD.PROFILE  -
  NEWNAME(XPED.OLD.PROFILE)
ALTER                -
  XPED.PROD.PROFILE.* -
  NEWNAME(XPED.OLD.PROFILE.*)

/* Change new dataset to production DSN */

ALTER                -
  XPED.NEW.PROFILE   -
  NEWNAME(XPED.PROD.PROFILE)
ALTER                -
  XPED.NEW.PROFILE.* -
  NEWNAME(XPED.PROD.PROFILE.*)
/*
/*

```

## Step 9. Review Global Parameters

Compuware recommends that your site install and test Xpediter/CICS with the shipped options. You may, however, want to review the global parameters in Chapter 5, “Global Parameters Table” in the *Xpediter/CICS Installation and Configuration Guide*.

### IMPORTANT:

Xpediter’s restricted modes are enabled by changing the default settings for the global table parameters OKUPDT and UTILMOD. See the *Xpediter/CICS Reference Manual* for more information on using restricted modes.

## Step 10. Create the Global Table Parameters Override Dataset (Optional)

Xpediter/CICS provides a facility for overriding global parameters with values read from a sequential dataset at product initialization. A standalone transaction, XSIT, can also be used to override global parameters from the same dataset while Xpediter is running.

Compuware strongly recommends using the override dataset method of overriding parameters because it provides greatly enhanced ease of use. To create the required dataset and enable the override facility, follow the instructions in Chapter 5, “Overriding Global Parameters” in the *Xpediter/CICS Installation and Configuration Guide*.

**Note:** If you have been using a global parameters override dataset, review the parameters for possible changes that may be required for your site. You may continue to use your existing dataset.

## Step 11. Allocate and Initialize the Recreate Dataset (DBUGUD) (Optional)

Xpediter/CICS includes a documentation packaging utility that can assist you, if necessary, in collecting and transmitting data to Xpediter/CICS Customer Support. That data allows Compuware to recreate your environment and resolve support issues more quickly.

**Note:** If you have been using the Recreate dataset, you may continue to use your existing dataset.

Unless the documentation packaging utility has been disabled by changing the value of global parameter RECREATE to NONE, either in DBUGGBL or as a global table override, you must use one of the following methods to allocate and initialize the Recreate dataset DBUGUD:

- Invoke the DOCINIT member of dataset SMXDSAMP member using the TSO EXECUTE command. For example:

```
TSO EX 'CPWR.dMXD940.SMXDSAMP(DOCINIT)'
```

DOCINIT is a REXX EXEC similar to the Installation JCL Customization Facility (XDINST). Be sure the values entered for DOCINIT are the same as those entered during installation.

- Use the JCL provided in member SMXDSAMP. Edit the JCL (shown in Figure 1-4) to your site’s standards, then submit it to allocate and initialize the Recreate dataset.

**Note:** This member contains control characters in columns 79 and 80. Do *not* edit with NUM option or modify these columns.

After completing the full installation, you have the option of contacting Xpediter/CICS Customer Support to arrange a test of the documentation packaging utility that will verify its proper configuration.

**Figure 1-4.** Sample JCL (DBCDEF RP) to Allocate and Initialize Recreate Dataset DBUGUD

```
//TEMPLATE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE XPED.RECREATE.JCL CL PURGE
DEFINE CLUSTER                                -
  (NAME(XPED.RECREATE.JCL)                   -
   VOLUMES(vvvvvv)                          -
   TRACKS(2 2)                               -
   RECORDSIZE(80 80)                        /* DO NOT MODIFY */ -
   NUMBERED                                  /* DO NOT MODIFY */ -
   SHAREOPTIONS(1 3))                       /* DO NOT MODIFY */ -
/*                                           LOAD JCL DATA          */
IF LASTCC EQ 0                               -
THEN REPRO                                    -
  INFILE(SYSJIN)                             -
  OUTDATASET(XPED.RECREATE.JCL)
/*
//SYSJIN DD DATA,DLM=@@
/* SEE MEMBER DBCDEF RP */
@@
//
```

## Step 12. Review Automatic Session Termination

Xpediter/CICS can be tailored to terminate an active debugging session automatically if a terminal is disconnected, logged off, and/or signed off from CICS. Refer to Chapter 9, “Session and Product Termination” in the *Xpediter/CICS Installation and Configuration Guide* for additional information on this option.

### Notes:

- Compuware highly recommends that you install automatic session termination to release the resources of sessions that have not been manually terminated.
- If you have been using Automatic Session Termination, you may continue to use your existing implementation.

## Step 13. Install and Customize for Xpediter/Eclipse (Optional)

Xpediter/Eclipse consists of an Eclipse plug-in that provides the ability to test and debug applications that execute in CICS and MVS environments. This step describes the actions necessary to enable a user running Eclipse on their workstation to interface with the Xpediter/CICS Sockets-Based Debugging API.

### Installation of Additional Components

Before proceeding, make sure the necessary components and plug-ins have been installed in accordance with the *Compuware Workbench Installation Guide*.

### Verify IBM z/OS Communications Server Installation

Use of Xpediter/Eclipse components requires your site to install IBM's z/OS Communications Server product. This facility provides the TCP/IP v4 support necessary to provide sockets communications between the various components of Xpediter/Eclipse, Xpediter/CICS, and the Compuware ECC for Eclipse component.

**Note:** Xpediter/Eclipse and its components do not contain support for TCP/IP v6 communications. Both TCP/IP v4 and v6 can coexist within the z/OS Communications Service.

Only the steps necessary to customize your current IP CICS Sockets are covered in this step. If you do not currently have IP CICS Sockets defined in your CICS regions, you must follow the installation steps outlined in the *IP CICS Sockets Guide* shipped with IBM's z/OS Communications Server.

Compuware strongly recommends the installation of the EZAC transaction and components shipped with IBM's z/OS Communications Server. Compuware also recommends that you automatically start and shut down the IP CICS Sockets interface according to the instructions provided in the *IP CICS Sockets Guide*, in the section entitled "Managing IP CICS Sockets: Starting and stopping CICS automatically".

## Define Transaction Code for Default IP CICS Sockets Listener

Xpediter/CICS requires use of the default CICS sockets listener, EZACIC02, shipped with the z/OS Communications Server.

Modify the sample RDO definition shown in Figure 1-5 as follows, then add it to any CICS region that an Xpediter/Eclipse user may connect to:

- Although you can choose a different transaction code, Compuware suggests you use XSKL. If you choose another transaction code, make note of it for later use in "Define Listener to IP CICS Sockets" on page 1-20.
- The program name in the definition below, EZACIC02, may need to be modified.
  - If your site has not created a customized version of EZACIC02, leave the PROGRAM parameter unchanged.
  - If your site is running a customized version of EZACIC02, determine the load module name of the customized version as follows:
    - If the customized version is not named EZACIC02, leave the PROGRAM parameter unchanged.
    - If the customized version is named EZACIC02, change the PROGRAM parameter to whatever name the original EZACIC02 was renamed to.
    - If the original EZACIC02 was not renamed, you must recover and rename it something other than EZACIC02, then use that new name for the PROGRAM parameter.
- Transaction XSKL, or the transaction code you decide to use as the equivalent, should not be secured. If you wish to limit access to Xpediter/CICS by use of Xpediter/Eclipse, you should instead secure transaction code XPGD as described in the next section.

**Figure 1-5.** RDO Transaction Definition for XSKL

```
DEFINE -
TRANSACTION(XSKL) -
DESCRIPTION('Xpediter/CICS Eclipse Listener Task') -
GROUP(XPEDR920) -
PROGRAM(EZACIC02) -
PRIORITY(255) -
TASKDATALOC(ANY) -
TASKDATAKEY(CICS)
```

## Define Transaction Code for Limiting Access to Xpediter/CICS Using Xpediter/Eclipse

Xpediter/CICS requires that you define transaction XPGD, or one of your choosing, as shown in Figure 1-6. This transaction can be used to limit access to Xpediter/CICS debugging facilities through the use of Xpediter/Eclipse. Even if you choose to not limit access, the transaction must be defined.

- If you want to use a different transaction code:
  - It must have the attributes shown in Figure 1-6.
  - You must specify the new transaction code using the Xpediter/CICS global parameter `XPGD=xxxx`, where `xxxx` is the transaction code you elect to use. For more information, see “XPGD” in Chapter 4 of the *Xpediter/CICS Installation and Configuration Guide*.
- If you do not wish to limit access, make sure you do **not** secure this transaction.
- If your site wants to limit access to Xpediter/CICS debugging facilities through the use of Xpediter/Eclipse, you may do so by limiting access to transaction XPGD. This transaction can be secured by defining user profiles in the ACICSPCT class. Any users added should have READ access. Only users who pass this security check will be allowed access to Xpediter/CICS debugging facilities using Xpediter/Eclipse.

**Figure 1-6.** RDO Transaction Definition for XPGD

```
DEFINE -
TRANSACTION(XPGD) -
DESCRIPTION(Xpediter/CICS Sockets Security) -
GROUP(XPEDR920) -
PROGRAM(XDPIMIRS) -
TASKDATALOC(ANY) -
TASKDATAKEY(CICS)
```

## Define Xpediter/CICS Sockets Child Listener/Interface

The Xpediter/CICS Sockets Child Listener/Interface program, DBUGSOCK, is a required component of the Xpediter/CICS Sockets-Based Debugging API. This program handles the TCP/IP sockets communication between the various components of the Xpediter/Eclipse package.

Add the RDO definition shown in Figure 1-7 to any CICS region that an Xpediter/Eclipse user may connect to:

**Figure 1-7.** RDO Program Definition for Xpediter/CICS DBUGSOCK

```
DEFINE
PROGRAM(DBUGSOCK)
DESCRIPTION('Xpediter/CICS Eclipse Sockets Interface')
GROUP(XPEDR920)
LANGUAGE(ASSEMBLER)
EXECCKEY(CICS)
DATALOCATION(ANY)
```

## Define Transaction Code for Xpediter/CICS Sockets Child Listener/Interface

The transaction code defined below ensures that the appropriate connections can occur between the various components of the Xpediter/Eclipse package and will be used when specifying CICS entries in the Xpediter TP Configuration file.

Although you can choose a different transaction code, Compuware suggests you use XSOC. If you choose another transaction code:

- Make note of it for later use in “Define Listener to IP CICS Sockets” on page 1-20.
- It must have the attributes shown in Figure 1-8.

Add the RDO definition shown in Figure 1-8 to any CICS region that an Xpediter/Eclipse user may connect to.

**Figure 1-8.** RDO Transaction Definition for XSOC

```
DEFINE
TRANSACTION(XSOC)
DESCRIPTION('Xpediter/CICS Sockets Child Listener/Interface')
GROUP(XPEDR920)
PROGRAM(DBUGSOCK)
PRIORITY(255)
TASKDATALOC(ANY)
TASKDATAKEY(CICS)
```

## Define Xpediter/CICS Sockets Security Exit

Xpediter/CICS provides a sockets security exit, XDPCICSE, which allows users of Xpediter/Eclipse to be authenticated. When establishing a debugging session from Xpediter/Eclipse, a user must provide their mainframe userID and password. This data is encrypted and passed during the connection process. Before the sockets listener attempts to pass control to the Xpediter/CICS Sockets Child Listener/Interface, additional checks are made and the userID and password are verified. The exit that does this is specified in the IP CICS Sockets Listener definition in the next section. If the userID/password is authenticated, IP CICS Sockets logs the user onto CICS and the task under which the Xpediter/Eclipse session is running is assigned that userID. If the userID/password cannot be authenticated, the task ends and access is not allowed.

The following program definition is required. If you wish to ensure userID/password authentication and logon, you must specify the parameter SECEXIT=XDPCICSE in the IP CICS Sockets Listener definition in the next section. If you do not require a user to log on to CICS, leave this parameter blank. This will cause the default CICS userID to be assigned to the task under which the Xpediter/Eclipse session runs.

Add the RDO definition shown in Figure 1-9 to any CICS region that an Xpediter/Eclipse user may connect to and have their userID authenticated:

**Figure 1-9.** RDO Program Definition for Xpediter/CICS Sockets Security Exit XDPCICSE

```
DEFINE
PROGRAM(XDPCICSE)
DESCRIPTION('Xpediter/CICS Sockets Security Module')
GROUP(XPEDR920)
LANGUAGE(ASSEMBLER)
EXECCKEY(CICS)
DATALOCATION(ANY)
```

## Define Listener to IP CICS Sockets

This step provides the definitions required to enable the Xpediter/Eclipse components to connect and communicate properly with the Xpediter/CICS Sockets-Based Debugging API. Many of the parameters listed below must be specified exactly as shown. Modifying them in any way can result in the unavailability of the Xpediter/CICS Sockets-Based Debugging API and complicate problem determination. If you are unsure of any

parameter, code it as shown in the example in Figure 1-10.

The definitions shown in Figure 1-10 have been formatted as though the parameters were being entered in your original EZACICFG. Although it is possible to add them to the EZACICFG JCL stream, doing so might cause other definitions to be lost. If you have experience with regenerating your configuration dataset, use the following definitions. However, Compuware strongly recommends that you utilize the existing EZAC transaction—which can be secured and limited to selected users—and enter them interactively using the EZAC DEFine and ALTer parameters.

**IMPORTANT:**

- If you are sharing a configuration dataset among several regions, make sure you create identical definitions for all regions you want Xpediter/Eclipse users to be able to connect with. Each definition would require you to modify the APPLID and PORT parameters at a minimum. If you decided above to modify the CICS RDO definition for the XSKL or XSOC transactions, or if you need to create unique definitions by region, you must modify the TRANID and CSTRAN parameters as well. Do **not** modify any underlined parameter settings.
- The USERID parameter can be assigned a default userID to be assigned to the task under which the Xpediter/CICS interface to Xpediter/Eclipse will run. You can choose to specify a value or not.
- The SECEXIT parameter can be used if you want to ensure that the user attempting to use Xpediter/Eclipse can be authenticated. If you want to require authentication, the program XDPCICSE must be specified. If a user cannot be authenticated, access to Xpediter/CICS debugging facilities through Xpediter/Eclipse will be prohibited.
  - If your site will be using Compuware’s Runtime Visualizer, you **must** specify XDPCICSE for the SECEXIT parameter.

**Note:** If the EZAC transaction is used, the SECEXIT parameter will be listed as **USEREXIT**.

- WLM (Work Load Manager) parameters have been explicitly omitted. Support for this facility will be implemented in a future release.

**Figure 1-10.** Definitions for Enabling Listener to IP CICS Sockets

EZACICD	<u>TYPE=LISTENER,</u>	Create Listener Record (XP Eclipse)	X
	<u>FORMAT=ENHANCED,</u>	Enhanced Listener	X
	APPLID=H01AC024,	APPLID of CICS	X
	TRANID=XSKL,	Use Clone of Standard Transaction ID	X
	PORT=nnnnn,	CICS Port Number of listener	X
	<u>AF=INET,</u>	IPv4	X
	<u>IMMED=YES,</u>	Start Listener Immediately	X
	BACKLOG=20,	Set Backlog Value to 20	X
	NUMSOCK=50,	Support 50 Concurrent Connections	X
	ACCTIME=30,	Set Timeout Value to 30 Seconds	X
	GIVTIME=30,	Set Givesocket Timeout to 30 Seconds	X
	REETIME=300,	Set Read Timeout to 5 minutes	X
	CSTRAN=XSOC,	TranID Associated with DEBUGSOCK	X
	<u>CSSTYP=KC,</u>	Child Server Startup Type	X
	<u>CSDELAY=000000,</u>	Child Server Delay Interval	X
	<u>MSGLEN=61,</u>	Length of Input Message	X
	<u>PEEKDAT=NO,</u>	Peek Option	X
	<u>MSGFORM=EBCDIC</u>	Output Message Format	X
	USERID=aaaaaaaa,	Default userid to be assigned	X
	SECEXIT=XDPCICSE	Security Exit	

## Enable Xpediter/CICS Sockets-Based Debugging API

Install the RDO definitions created above by either submitting CEDA transaction to specify your definitions or by creating sequential input versions of the definitions and adding them with the DFHCSDUP batch utility.

After the RDO definitions have been installed, activate the new listeners using the EZAO transaction. This is not necessary, however, if you have stopped and started IP CICS Sockets.

## Test Xpediter/Eclipse Connection

After successfully completing the above steps and checking your CICS and JES logs for errors, test your Xpediter/Eclipse connection. Make sure your Xpediter TP Configuration File member contains valid CICS entries that point to any CICS regions you want Xpediter/Eclipse users to connect to. For more information, see the section entitled “Part 4: Create and Manage the CSS TP Configuration File” in Chapter 3 of the *Compuware Workbench Installation Guide*. You should also make sure that your Eclipse Debug Launch Configuration settings have been properly specified and you can connect to the Xpediter/Eclipse TP. If you are unable to connect, double-check the installation of the Xpediter/Eclipse components and review any messages written to the JES and CICS logs to assist in verifying their correct installation.

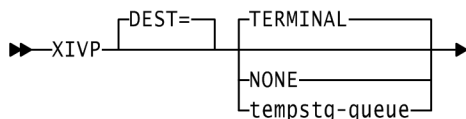
## Additional Information

A new Xpediter/CICS Global Override, XPGD=xxxx, has been added for supporting the Xpediter/Eclipse interface. This override allows the default transaction id XPGD to be renamed to one of your choosing (xxxx). If you decide to modify this parameter, you must also ensure that its definition matches that shown in Figure 1-6 in the section “Define Transaction Code for Limiting Access to Xpediter/CICS Using Xpediter/Eclipse” above.

Several new messages with the prefix MXDSK have been added to support the Xpediter/Eclipse interface. The Xpediter/CICS Sockets modules XDPCICSE and DBUGSOCK output these messages. They provide informational data, and many of them are related to the setup of the Xpediter/Eclipse interface. You are advised to review the CSMT output during and after installation or if connection or usage problems occur while attempting to use the Xpediter/Eclipse interface.

## Step 14. Verify the Xpediter/CICS Installation

To verify that the Xpediter/CICS installation has been successful, Xpediter/CICS runs the Installation Verification Process (IVP). This procedure, performed automatically when the product is initialized in the CICS region, detects any changes introduced in the region that would cause the product not to function. You may request verification by using the XIVP transaction. This transaction routes the resulting output messages to the destination of your choice. The syntax for XIVP is as follows:



- |                           |  |
|---------------------------|--|
| <b>DEST=TERMINAL</b>      | Causes the output to go to the terminal at which XIVP is entered. This is the default. |
| <b>DEST=NONE</b>          | Causes all output to go only to transient data destination CSMT.                       |
| <b>DEST=tempstg-queue</b> | Causes the output to go to the temporary storage queue name supplied.                  |



The global table parameter IVPMSGs controls the destination of messages. To change this parameter, refer to Chapter 4, “Global Parameters Table” in the *Xpediter/CICS Installation and Configuration Guide*.

All discrepancies detected by XIVP are assigned a severity level of SEVERE, ERROR, WARNING or INFORMATIONAL.

If severe errors are detected during initialization, they are placed in the temporary storage queue XPEDXIVP, and Xpediter/CICS abends with a code of DBI0. Refer to Appendix A in the *Xpediter/CICS Reference Manual* for information on the abend code. If the product is being initialized at a terminal where errors are detected, the following messages are directed to the terminal user before the abend is issued:

```

ERRORS WERE DETECTED DURING INITIALIZATION
USE CEBR TO BROWSE TEMPSTG QUEUE XPEDXIVP, OR
EXAMINE TRANSDATA DESTINATION CSMT FOR MESSAGES, OR
EXECUTE TRANSACTION XIVP DEST=TERMINAL TO SEE ERRORS

```

If ERROR level discrepancies are detected during initialization, they are placed in the temporary storage queue XPEDXIVP, and Xpediter/CICS initialization will continue. If the product is being initialized at a terminal, the messages shown below will be directed to the terminal user. If the user responds **NO** to the “continue initialization” prompt, Xpediter/CICS abends with DBI0.

```

                ERRORS DETECTED

THE PRODUCT MAY BE USED, BUT UNPREDICTABLE RESULTS
MAY OCCUR, UP TO AND INCLUDING POSSIBLE REGION
OUTAGES.  CORRECT THESE ERRORS AS SOON AS POSSIBLE.

CONTINUE INITIALIZATION? ==>          (YES/NO)

```

If product is initialized with ERROR level discrepancies, a message “Xpediter initialization error, execute XIVP” will be issued to the terminal when a new Xpediter/CICS session is started. This will alert each user to the fact that there may be problems with the product’s configuration. These errors should be corrected as soon as possible.

