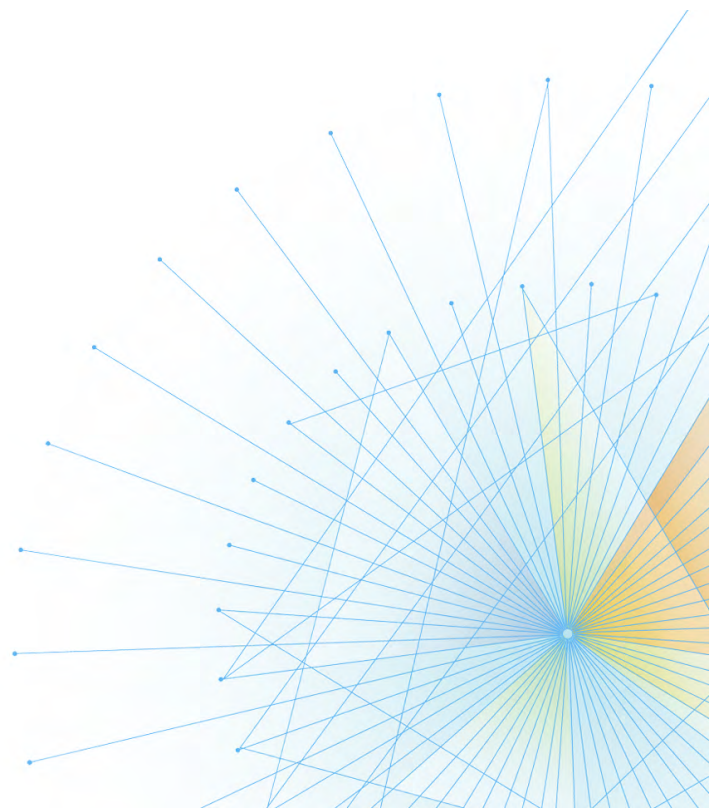




The Mainframe Software Partner For The Next 50 Years

# Xpediter/*Xchange* Installation and Configuration Guide

**Release 17.02**



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

**Compuware Support Center**

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

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

Copyright 1995 - 2019 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 Unpublished rights reserved under the Copyright Laws of the United States.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.

*Xchange* and Xpediter are trademarks or registered trademarks of Compuware Corporation.

CICS, CICS TS, DB2, IBM, IMS, MVS, MVS/ESA, OS/390, RACF, VTAM, 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.

# Contents

<b>Introduction</b> .....	<b>5</b>
Overview .....	5
Icons .....	5
<b>Xpediter/Xchange Overview</b> .....	<b>7</b>
What is Xpediter/Xchange? .....	7
Features .....	7
<b>Planning</b> .....	<b>9</b>
Steps Involved .....	9
Milestones and Roles .....	9
Packaging, Libraries, and Maintenance .....	10
Preinstallation Considerations .....	11
Checklist of Milestones and Tasks .....	17
System Environment .....	17
<b>Milestone 1: Ensure Installation and Configuration of Companion Products</b> .....	<b>19</b>
<b>Milestone 2: Install Xpediter/Xchange Using SMP/E</b> .....	<b>21</b>
<b>Milestone 3: Specify Xpediter/Xchange Properties and Parameters</b> .....	<b>23</b>
<b>Milestone 4: Update TSO CLISTs</b> .....	<b>29</b>
<b>Milestone 5: Allocate Journal and Save/Restore Datasets</b> .....	<b>31</b>
<b>Milestone 6: Enable Language Environment (LE) Support</b> .....	<b>35</b>
<b>Milestone 7: Enable Xchange SVCs</b> .....	<b>39</b>
Starting Xchange as a Started Task .....	39
Tasks .....	39
<b>Milestone 8: Configure Xchange External Security</b> .....	<b>41</b>
Overview .....	41
<b>Milestone 9: Assemble or Compile and Link Edit Test Programs</b> .....	<b>47</b>
<b>Milestone 10: Enable CICS Support</b> .....	<b>51</b>
<b>Milestone 11: Enable DB2 Support</b> .....	<b>55</b>
Overview .....	55
<b>Milestone 12: Enable IMS Support</b> .....	<b>57</b>
<b>Milestone 13: Enable Xpediter/TSO Interface</b> .....	<b>61</b>
<b>Milestone 14: Enable Natural Support</b> .....	<b>63</b>
<b>Milestone 15: Enable COPE Logical System Support</b> .....	<b>65</b>
<b>Milestone 16: Deployment</b> .....	<b>67</b>

**Checklist of Milestones and Tasks ..... 69**

# Introduction

This manual provides information about how to install, customize, and maintain Xpediter/*Xchange*.

## Overview

This document is intended to guide you through installing/updating, configuring, deploying, and troubleshooting Xpediter/*Xchange*. Supplemental documentation can be found in the *Xpediter/Xchange User Guide*.

## Icons

The icons found in this guide include:



**“Tip”** A note or tip providing additional information.



**“Remember”** Information important to remember.



**“Fast Forward”** If a particular milestone or task doesn’t apply to your site—or your site is not licensed for a particular option—you can skip ahead to the next milestone or task by clicking the icon.



**“Roles”** The individuals required to perform a Milestone or Task.



**“Caution”** Failure to follow these instructions can cause problems.



# Xpediter/*Xchange* Overview

With all the date and time dependencies in modern applications—even after the year 2000 has come and gone—many in the data processing community still face a distinct challenge. Even programs with a four digit year in their date format (MM/DD/YYYY) require time-dependent testing to verify functions such as month-end and leap year processing. Inadequate and untested code can lead to invalid results for comparison, subtraction, division, and sorting routines.

Because so many application programs have required so many modifications over the past few years, an incredible amount of testing needs to be done. But how do you simulate a specific date and time for some programs without affecting all the rest? Compuware's Xpediter/*Xchange* serves as the perfect tool to automate application date and time testing.

## What is Xpediter/*Xchange*?

Xpediter/*Xchange* is a sophisticated tool for moving your time-sensitive applications into the year 2000 and beyond. By using *Xchange*, you'll be prepared to address the potentially costly impact that improper date and time manipulations can have on your applications.

*Xchange* automates date and time testing in your applications. You can build an inventory of all applications that make system date and time requests, and then test those applications using different dates and times.

You can simulate century date changes, month-end and leap year processing, and a variety of other exception conditions on a job-by-job, program-by-program, or application group basis. Use *Xchange* to test different dates for any program without dedicating a separate system or affecting other jobs that are running simultaneously.

You can set the values for the date and time you want to simulate. Then, *Xchange* will substitute, or exchange, this value each time the programs you specify request the system date and time with a TIME macro, EXEC CICS ASKTIME, STCK, STCKE, or STCKF instruction, or DB2 call. You can tell by looking at the results of your application whether it handled the date change in an appropriate manner. If not, just make the necessary changes and test again using *Xchange*.

## Features

In addition to the new release features described in the *Xpediter/Xchange Release Notes*, *Xchange* continues to include the following features:

- **Extended Batch Facility.** By submitting a stand-alone job, you can add, delete, advance, and list *Xchange* requests from outside of the ISPF, CICS, and IMS interfaces.
- **COPE Requests and Exclusions.** Now you can exchange dates and times for all jobs running under a specified COPE Logical IMS System. A new *Xchange* parameter allows you to designate the logical systems under which date and time exchanges are to be allowed.
- **Save and Restore of *Xchange* Requests.** All your date and time simulation requests are now saved at user-specified intervals and automatically restored whenever *Xchange* is restarted.
- **Jobclass Requests and Exclusions.** Now you can exchange dates and times for all jobs running under a specified jobclass. A new *Xchange* parameter allows you to designate the jobclasses under which date and time exchanges are to be allowed.

- **UserID and Asynchronous Task CICS Requests.** *Xchange* requests can now be set by CICS userID, and a single request can be used to specify both userID and asynchronous tasks.
- **ISPF Interface.** *Xchange's* ISPF interface lets you enter time simulation requests for jobs, steps, procedure steps, and programs. The interface consists of a screen that allows you specify individual entries or use a wildcard feature to specify multiple jobs, steps, procedure steps, and program names.
- **Batch Facility.** By inserting an *Xchange* batch step in your JCL, you can modify system date and time requests in batch for your application.
- **Xpediter/TSO Interface.** Users of Xpediter/TSO can dynamically request *Xchange* to set a simulated date and time for programs they are testing. *Xchange* is simply invoked from the COMMAND line of the Xpediter/TSO Source screen, allowing thorough and immediate testing of the program's date and time logic.
- **Xchange Journal.** *Xchange's* journal provides a comprehensive record of system date and time requests that have been exchanged. You can access the journal online or print an activity report.
- **Security.** *Xchange* works within the confines of your existing external security system to ensure that unauthorized access is not granted to any programs. Before an *Xchange* request is activated, external security validates the authorization of the request originator.
- **CICS Transaction Support.** *Xchange* uses its CICS transaction feature to exchange the date and time for individual programs, terminals, and transactions within a CICS region.
- **Language Environment Support.** *Xchange* fully supports LE's common execution environment for the following IBM high-level language (HLL) products:
  - Enterprise COBOL
  - COBOL/370
  - COBOL for MVS & VM
  - PL/I for MVS & VM
  - Enterprise PL/I
  - C/370, C/C++ MVS (excluding native language services), and z/OS XL C
  - Language Environment-conforming Assembler.
- **DB2 Support.** Dates and times returned by DB2 calls are exchanged. *Xchange* intercepts every DB2 date or time request but changes the date or time only when a match is found with one of your *Xchange* requests.
- **IMS Message Level Support.** *Xchange* uses its IMS Message Level feature to exchange the date and time for individual programs within an IMS message region.
- **Support for LINKAGE=SYS.** In addition to the LINKAGE=SVC parameter, *Xchange* also supports the TIME macro parameter LINKAGE=SYS.
- **STCK Support.** *Xchange* supports the Store Clock (STCK) operation code used by many Assembler programs to store the current value of the CPU time-of-day clock. This support is augmented by Enhanced STCK Support beginning with Release 3.1.
- **STCKE Support.** *Xchange* supports the Store Clock Extended (STCKE) operation code used by newer Assembler programs to store the current value of the extended CPU time-of-day clock. This support is augmented by Enhanced STCKE Support.
- **STCKF Support.** *Xchange* supports the Store Clock Fast (STCKF) operation code used by some Assembler programs to store the current value of the CPU time-of-day clock. This support is augmented by Enhanced STCKF Support.
- **Journal Exit.** When a journal record is written for a date or time exchange, *Xchange* can invoke a journal exit that determines the program ID to be reported.



# Planning

This section provides information related to planning to install or update to Xpediter/*Xchange* 17.02.

## Steps Involved

1. Order Xpediter/*Xchange* and its companion products, including the latest maintenance, via Compuware's Product Ordering web page or by contacting Compuware Customer Solutions.
2. Read this *Installation and Configuration Guide* and complete each of the milestones to:
  - a. Ensure companion products have been installed and configured with the latest maintenance and that the license for Xpediter/*Xchange* has been imported.
  - b. Perform the SMP/E installation of Xpediter/*Xchange* according to the *Compuware Installer Mainframe Products SMP/E Installation Guide*.
  - c. Implement the Compuware Mainframe Services Controller (CMSC) PARMLIB.
  - d. Configure Xpediter/*Xchange* for either a new installation or an upgrade.
  - e. Perform additional configuration for other features.
  - f. Deploy Xpediter/*Xchange* to additional environments.

## Milestones and Roles

Installation, configuration, verification, and deployment are done in 16 milestones. The rows in [Table 1](#) identify the role or skill set required to perform each milestone. This makes it easier to know which people need to be involved at each milestone along the way. With the proper planning, you may be able to have certain tasks performed at the same time.

**Table 1.** People Required for Each Milestone

Milestone	Companion Product Installer/Administrator	Xpediter/ <i>Xchange</i> Installer	z/OS System Programmer	z/OS Security Administrator	VTAM Administrator	DBA
<a href="#">Milestone 1: Ensure Installation and Configuration of Companion Products</a>	ECC ●					
<a href="#">Milestone 2: Install Xpediter/<i>Xchange</i> Using SMP/E</a>		●				
<a href="#">Milestone 3: Specify Xpediter/<i>Xchange</i> Properties and Parameters</a>		●				
<a href="#">Milestone 4: Update TSO CLISTs</a>		●				
<a href="#">Milestone 5: Allocate Journal and Save/Restore Datasets</a>		●				
<a href="#">Milestone 6: Enable Language Environment (LE) Support</a>		●				

**Table 1.** People Required for Each Milestone (*Continued*)

Milestone	Companion Product Installer/Administrator	Xpediter/Xchange Installer	z/OS System Programmer	z/OS Security Administrator	VTAM Administrator	DBA
<a href="#">Milestone 7: Enable Xchange SVCs</a>		●	●			
<a href="#">Milestone 8: Configure Xchange External Security</a>		●		●		
<a href="#">Milestone 10: Enable CICS Support</a>		●	●		●	
<a href="#">Milestone 11: Enable DB2 Support</a>		●	●			DB2 ●
<a href="#">Milestone 12: Enable IMS Support</a>		●				IMS ●
<a href="#">Milestone 13: Enable Xpediter/TSO Interface</a>		●				
<a href="#">Milestone 14: Enable Natural Support</a>		●				
<a href="#">Milestone 15: Enable COPE Logical System Support</a>		●				IMS ●
<a href="#">Milestone 16: Deployment</a>		●				

## Packaging, Libraries, and Maintenance

### Xpediter/Xchange Packaging

Compuware has registered the following element prefixes with IBM for the Xpediter/Xchange components:

- LXG for Xpediter/Xchange
- KCW as a generic Compuware prefix

The FMID for Xpediter/Xchange is MLXG $nnn$ , where  $nnn$  is the release number code, for example MLXG170.

### Libraries Created During SMP/E Installation

[Table 2](#) lists the libraries created during the Xpediter/Xchange installation using SMP/E, where  $nnn$  is the release number code, for example CPWR.MLXG170.DZONE.CSI.

**Table 2.** Libraries Created During Installation with SMP/E

ddname	Library Type and Content	Dataset Name as Distributed
SMPCSI	Compuware Global CSI	CPWR.GLOBAL.CSI
SMPLOG	Compuware SMP/E System File	CPWR.GLOBAL.SMPLOG
SMPPTS	Compuware SMP/E System File	CPWR.GLOBAL.SMPPTS
MXG $nnn$ D	Base Distribution Zone	CPWR.MLXG $nnn$ .DZONE.CSI
MXG $nnn$ T	Base Target Zone	CPWR.MLXG $nnn$ .TZONE.CSI
SMPTLIB	Base SMP/E System File	CPWR.MLXG $nnn$ .F1
SMP#####	Base SMP/E System File	CPWR.MLXG $nnn$ .F2
SMP#####	Base SMP/E System File	CPWR.MLXG $nnn$ .F3

**Table 2.** Libraries Created During Installation with SMP/E (*Continued*)

ddname	Library Type and Content	Dataset Name as Distributed
SMPMTS	Base SMP/E System File	CPWR.MLXGnnn.SMPMTS
SMPSCDS	Base SMP/E System File	CPWR.MLXGnnn.SMPSCDS
SMPSTS	Base SMP/E System File	CPWR.MLXGnnn.SMPSTS
ALXGCLIB	Base Distribution Clist Library	CPWR.MLXGnnn.ALXGCLIB
ALXGCNTL	Base Distribution CNTL Library	CPWR.MLXGnnn.ALXGCNTL
ALXGDATA	Base Distribution Data Library	CPWR.MLXGnnn.ALXGDATA
ALXGLOAD	Base Distribution Load Library	CPWR.MLXGnnn.ALXGLOAD
ALXGMENU	Base Distribution Message Library	CPWR.MLXGnnn.ALXGMENU
ALXGPENU	Base Distribution Panel Library	CPWR.MLXGnnn.ALXGPENU
ALXGSAMP	Base Distribution Sample Library	CPWR.MLXGnnn.ALXGSAMP
SLXGAUTH	Base Target Authorized Loadlib	CPWR.MLXGnnn.SLXGAUTH
SLXGCLIB	Base Target Clist Library	CPWR.MLXGnnn.SLXGCLIB
SLXGCNTL	Base Target CNTL Library	CPWR.MLXGnnn.SLXGCNTL
SLXGDATA	Base Target Data Library	CPWR.MLXGnnn.SLXGDATA
SLXGLOAD	Base Target Loadlib	CPWR.MLXGnnn.SLXGLOAD
SLXGMENU	Base Target Message Library	CPWR.MLXGnnn.SLXGMENU
SLXGPENU	Base Target Panel Library	CPWR.MLXGnnn.SLXGPENU
SLXGSAMP	Base Target Sample Library	CPWR.MLXGnnn.SLXGSAMP

## SMP/E Maintenance

Compuware recommends that you periodically obtain and apply maintenance to keep your product version current. All maintenance for Xpediter/*Xchange* and each of its releases is provided on the Compuware Support Center website at <http://go.compuware.com>.

Locate the **Fixes and Downloads** section for the Xpediter/*Xchange* product and select the release for which you wish to apply maintenance. Follow the instructions on the website for acquiring the maintenance.

Optionally you may acquire maintenance by getting a current copy of the Xpediter EP media and using it to generate the JCL for applying maintenance via SMP/E.

To install the Xpediter/*Xchange* maintenance to an existing installation, perform the steps described in Chapter 3, “Apply Maintenance with SMP/E” of the *Compuware Installer Mainframe Products SMP/E Installation Guide*.

## Preinstallation Considerations

This section includes important information that must be considered before beginning the *Xchange* installation procedure. Specifically, you must:

- [Ensure Specific MVS System Commands Run](#)
- [Collect Site-specific System Information](#)
- [Determine DASD Space Requirements](#)
- [Review Upgrade Considerations](#)
- [Designate an APF Authorized Library](#)
- [Consider How NUM Options Affect \*Xchange\* Use.](#)

Each of these preinstallation considerations are described below.

## Ensure Specific MVS System Commands Run

For the *Xchange* subsystem to properly detect job starts and completions, your site must run with the MVS system commands MONITOR JOBNames and MONITOR SESS. Consult your site's MVS system programmer.

## Collect Site-specific System Information

Collect the following information before you begin to install any component of *Xchange*:

- Generic names assigned to temporary units and DASDs (for example, DISK, SYSDA, VIO, or 3390).
- Unique four-character name you choose to assign to the *Xchange* subsystem. This value must be unique within the entire MVS complex. The subsystem ID used for a previous *Xchange* installation can be reused by performing one of the two options mentioned in [Review Upgrade Considerations](#) on page 12. The default is CWXG.
- Job cards to submit batch jobs.
- VSAM dataset requirements for special high-level qualifiers or separate volumes.
- External security high-level name qualifier.
- Values to specify for the NUM variables (see [Consider How NUM Options Affect Xchange Use](#) on page 13).
- *Xchange* requires an Authorized Program Facility (APF) library.

## Determine DASD Space Requirements

To calculate required disc space, refer to the allocations present in the jobs described in the *Compuware Installer Mainframe Products SMP/E Installation Guide*.

## Review Upgrade Considerations

This section lists various considerations related to upgrading from a previous release of *Xpediter/Xchange*.

- If you use the same MVS subsystem name for this release of *Xchange* as you used for the prior release and you have not IPLed your MVS image since you last started your prior release of *Xchange*, two options are available to you:
  - Run the XGSUBRST utility as documented in Chapter 6. This is the preferred method of upgrade as it eliminates the need for the MVS IPL. See the chapter entitled “Using the *Xchange* Utilities” in the *Xpediter/Xchange Installation and Configuration Guide* for more details.
  - IPL your MVS image.

If you use different subsystem names for the new release and the older one, you will be able to run both releases at the same time.

- If you plan to run both your currently installed release and the latest release of *Xchange*, then in [Specify Xpediter/Xchange Parameters](#) on page 24, keep in mind that:
  - The values for NUMAMB, NUMATB, PREVDATA, and other variables in your previous release can be used as initial values for this release.
  - If you want to run both your currently installed release and the latest release at the same time, the new release must have a different value specified for SUBSYS.

- If your security requirements are the same as with the previous release, you can use the same value for SECPFX.

If you will not be running two releases at the same time, you can reuse your *Xchange* journal datasets by specifying the same values on the LOGxxxx parameters.

- If you will be running a previous release and the most current release of *Xchange* at the same time, do the following:
  - In [Milestone 4: Update TSO CLISTS](#), provide separate TSO CLISTS for each *Xchange* release.
  - In [Milestone 8: Configure Xchange External Security](#), create new security rules if a new value was used for the SECPFX parameter.
  - In [Milestone 4: Update TSO CLISTS](#), provide separate Xpediter/TSO CLISTS for each *Xchange* release.
- If any changes were made to the source code in your previous release's RACROUTE module, then in [Milestone 8: Configure Xchange External Security](#) make the same changes to your new RACROUTE module.
- If your site requires IMS Message Level Support, you must perform [Enable IMS Message Level Support](#) on page 57, sub[Step 6](#) for XGTM (DATA member XGTMTM) and XGTS (DATA member XGTMTS).
- If your site requires CICS transaction support, you must use Resource Definition Online (RDO) to update your site's CICS tables with CEDA transactions as described in [Milestone 10: Enable CICS Support](#).

## Designate an APF Authorized Library

The *Xchange* address space must be executed from an Authorized Program Facility (APF) load library because it replaces the TIME services vectors and creates a new subsystem. All *Xchange* modules must be link-edited into this library, with the exception of those used for the ISPF, CICS, and batch interfaces, which do not need to run in an authorized state.

Prior to MVS/ESA release 4.3, the SYS1.PARMLIB member IEAAPFxx identified authorized libraries in a static list that could be updated only at IPL and could contain up to 253 entries. Since MVS/ESA release 4.3, the SYS1.PARMLIB member PROGxx can be used to identify authorized libraries in a static or dynamic list. When PROGxx uses a dynamic list, an unlimited number of authorized libraries can be specified, and the list can be updated without an IPL.

To use PROGxx:

1. Remove the APF=xx parameters from the IEASYSxx and IEASYS00 members of SYS1.PARMLIB.
2. Activate the PROGxx member with the SET PROG=xx operator command.
3. Define the *Xchange* APF authorized library, CPWR.MLXGmmn.SLXGAUTH, to MVS by adding the DSN and volume to IEAAPFxx or PROGxx (depending on MVS release) in SYS1.PARMLIB.

For more information about using PROGxx, consult your site's MVS system programmer.

## Consider How NUM Options Affect Xchange Use

Before you specify the options in the *Xchange* CMSC PARMLIB member, you need to know how the NUM variables (NUMAMB, NUMATB, NUMJRE, NUMORE, NUMURE, and NUMXRE) affect *Xchange* use. Discuss the intended use of *Xchange* with your site's users to determine the optimal values for the NUM variables and LOGxxxx parameters.

[Table 3](#) shows each NUM variable, its initialized (default) value, and the size in bytes of each occurrence of the specified variable. The total storage required in the Extended Common Storage Area

(ECSA) for each NUM variable can be determined by multiplying initialized value by the size in bytes of that variable.

**Table 3.** NUM Space Requirements

NUM Variable	Initialized Value	Size in Bytes
NUMAMB	128	326
NUMATB	256	448
NUMJRE	128	16
NUMORE	16	432
NUMURE	32	256
NUMXRE	64	344

If you expect to make a single subsystem available to a large number of users, Compuware recommends increasing the NUMAMB (number of jobs) value and the NUMATB (number of steps) value. Without the increases, some users might be unable to work with *Xchange*.

Each time you add 1 to NUMAMB increases the ECSA size by 326 bytes, and each time you add 1 to NUMATB increases the ECSA by 448 bytes. Because most jobs have multiple steps, you should normally increase NUMATB by 2 for every 1 that you increase NUMAMB. Therefore, increasing NUMAMB by 1 would require increasing the module by over 1K: 326 bytes for NUMAMB plus 896 bytes (448 x 2) for NUMATB.

If you change the default values for NUMAMB and NUMATB, a good starting number to use equals the maximum number of jobs that will be exchanged at any one time for NUMAMB. For NUMATB, use the NUMAMB value multiplied by your average number of steps per job.

Your site may choose to use multiple subsystems if, for example, you need to establish more restrictive security in production than you have in your test environment. If you will be using multiple subsystems, you should reduce the NUM variable values for each subsystem in order to reduce the overall resource consumption for *Xchange*. The other NUM parameters consume less resources than NUMAMB and NUMATB. NUMXRE, at 16 bytes, is unlikely to require considerable resources due to its small size.

When determining an appropriate NUMURE value, remember that it is directly affected by the *Xchange* workload, as identified by the number of requests entered via the ISPF interface in conjunction with the number of *Xchange* requests running on the system. In addition, setting the correct priority level for the *Xchange* address space can significantly enhance the throughput of the *Xchange* system by enabling rapid processing and resetting of the URE control blocks so they can be reused immediately. If *Xchange* has no opportunity to process the URE control blocks, they can accumulate until the NUMURE value is reached and exchanges no longer occur. See [Update the MVS Program Properties Table](#) on page 23 for information about setting the priority level. After *Xchange* has been given the proper priority level, you can make minor adjustments to the NUMURE value to meet your maximum workload needs.

NUMORE and NUMJRE represent queued units of work, which should be reset so quickly that their values need not be very high. As with NUMURE, resetting of the associated control blocks to enable immediate reuse is directly affected by the priority level of the *Xchange* address space.

## PARMLIB Members

Compuware mainframe products use parameter libraries, or PARMLIBs, to configure each product and common components. [Table 5](#) lists the Release 17.02 PARMLIB Members for all Compuware mainframe products and common components. Your product requires a PARMLIB where its members are provided such that you can modify them to fit your site's requirements, as described in [Implement the Compuware PARMLIB](#) on page 23.

Compuware recommends using one common dataset, but can be concatenation of dataset names to store your site's PARMLIB (if applicable) members in a common library. A copy of your product's parameter file(s) must reside in the //CWPARM DD **concatenation of the Compuware Mainframe Services Controller (CMSC)**.



See the *Enterprise Common Components Installation and Configuration Guide* release 17.02 for further information on CMSC.

The Compuware Mainframe Services Controller (CMSC) address space is a centralized facility providing the common parameter library services. It provides two basic functions in relation to parameters: storage and retrieval. The following is a guideline for some of the usage and features of the CMSC:

- Users will modify a human readable PARMLIB member and issue a modify command to the CMSC to store this new set of parameters into a common memory object.
- This common memory object is accessible even if the CMSC is inactive.
- When the CMSC is initialized, all PARMLIB members are loaded into a common memory object.
- The default suffix is 00, but this can be changed in the CMSCnnnn PARMLIB member.

## General Guidelines for PARMLIB Members

Parameters used by your Compuware mainframe product or component are read from the Compuware PARMLIB dataset. Edit the sample parameters to your site's requirements.

**Table 4.** General Guidelines

PARM Name	Values	Samples
Data columns	1 to 71	N/A
Symbol separator	Underscore	"A_LONG_SYMBOL"
Continuation	Check for "+" character in Column 72.	+
Comments	"*" in column 1. Between columns 1-71, begin with "/*" and end with "*/"	* is a comment /* is a comment */

- If the PARMLIB member includes multiple groups of parameters, for example for the definition of multiple DB2 Subsystems, then only one occurrence of each of the parameters within each group is allowed.
- Line level comments are supported using the "/\*" to start a comment and "\*/" to end the comment. Embedded comments are supported.

## System Symbolics

The CMSC resolves system symbolics as product parameters which are saved into storage, potentially allowing a single parameter member to be used across multiple LPARs. These symbols are defined by your installation in IEASYMxx. Issue the following display command to display the current symbols:

```
/DISPLAY SYMBOLS
```

## PARMLIB Member Naming Convention

PPPPnnnn

PPPP is the product prefix (see [Table 5](#)).

nnnn is the 1 to 4-character PARMLIB suffix, for example 00.

The PARMLIB member name for each product or product component must start with the product prefix, for example FACM for File-AID Common Components. The 1- to 4-character suffixes can be used to replace the default name (FACM00). If the parameter for a given product is omitted, the default suffix will be 00.

### Changing the Default PARMLIB Member

In the CMSC startup parameters, specify the product parameter, followed by the equal sign, and the 1- to 4-character suffix (for example: FACM=01, which points to PARMLIB member FACM01). If you changed your PARMLIB member name from the default FACM00 to FACM0005, for example, update your CMSC PARMLIB member to point to the new PARMLIB member, FACM=0005.

## General Guidelines for PARMLIB Dataset

- The dataset can be blocked.
- The dataset must be defined as Fixed Block.
- The dataset can have multiple extents.
- The dataset must be on a single volume.
- The CMSC must have READ access.

**Table 5.** Compuware Product PARMLIB Member Parameters

Product PARMLIB Member Parameters	Product
AABD	Abend-AID BDCAS
AADC	Abend-AID DCAS
AAFA	Abend-AID Fault Analytics
AATD	Abend-AID TDCAS
AAVW	Abend-AID Viewer
CMSC	Compuware Mainframe Service Controller
FACM	File-AID Common Component
FADA	File-AID/Data Solutions
FADE	File-AID DB2 Environment Information
FAMV	File-AID/MVS
FAFR	File-AID/RDX
FAFD	File-AID for DB2
FAIE	File-AID IMS Environment Information
FAIX	File-AID for IMS
HCI	Host Communications Interface
HSCM	Hiperstation
LMCL	License Management Client (LMSINIT)
LMSV	License Management Server (LMZINIT)
STR	Strobe
XCOV	Xpediter/Code Coverage
XVGB	Xpediter/Code Coverage CICS components
XDGB	Xpediter/CICS Global components
XDDB	Xpediter/CICS DBPA components
XTSO	Xpediter/TSO
XCHG	Xpediter/Xchange



## Necessary Information and Who to Ask

[Table 6](#) lists the information needed and who would be expected to provide it. You may want to print the table, then fill in your site's information to refer to during the installation process.

**Table 6.** Information Gathering Worksheet

Who to Contact	What is Needed	Your Information
<b>Required</b>		
<i>Xchange</i> SMP/E Installer	High-level qualifiers for <i>Xpediter/Xchange</i> SMP/E datasets	
ECC SMP/E Installer	Enterprise Common Components (ECC) SLCXLOAD dataset name	
ECC Administrator	Compuware Mainframe Services Controller (CMSC) PARMLIB dataset name	
ECC Administrator	Suffix defined/defaulted for <i>Xpediter/Xchange</i> in the CMSC	
<b>Required for Upgrade</b>		
Previous <i>Xchange</i> Installer	SLXGTABL dataset name from current <i>Xpediter/Xchange</i> installation	
<b>Required for IMS/DB</b>		
IMS System Programmer	IMS RESLIB (SDFSRESL) dataset name	
IMS System Programmer	IMS DBD library dataset name	
IMS System Programmer	IMS PSB library dataset name	

## Checklist of Milestones and Tasks

To keep track of your progress, you may want to print the [Checklist of Milestones and Tasks](#) at the end of this manual, then check off each Milestone and task as it gets completed.

## System Environment

### Software Supported

*Xpediter/Xchange* Release 17.02 supports the following environments:

- IBM z/OS V2.2, 2.3
- IBM ISPF for the supported z/OS releases
- IBM CICS Transaction Server for z/OS V5.1, 5.2, 5.3, 5.4, 5.5
- IBM IMS Transaction and Database Servers V14.1, 15.1
- IBM DB2 for z/OS V11.1, 12.1

### Languages

*Xchange* supports any language that uses a TIME macro, EXEC CICS ASKTIME, EIB date and time, or a DB2 call to retrieve date and time requests. Some of the supported languages are listed below.

#### Under IBM Language Environment for Supported z/OS Releases

- IBM Enterprise COBOL for z/OS V4.2, 5.1, 5.1.1, 5.2, 6.1, 6.2
- IBM Enterprise PL/I for z/OS V4.5, 5.1, 5.2
- IBM XL C/C++ for supported z/OS releases. Note: Support is limited to C, not C++.

### Non-LE Languages

- IBM High Level Assembler (including STCK, STCKE, and STCKF instructions when replaced with *Xchange* macro).

# Milestone 1: Ensure Installation and Configuration of Companion Products

Complete the following task to install and configure Xpediter/*Xchange* companion products.



Roles involved:  
ECC Installer

## Task 1.1 Install/Upgrade Enterprise Common Components

Enterprise Common Components (ECC) version 17.02 or higher must be installed and configured to support Xpediter/*Xchange* 17.02. See the *Enterprise Common Components Installation and Configuration Guide* for instructions on configuring ECC for use with Xpediter/*Xchange*.

## Task 1.2 Apply ECC Maintenance

Apply the latest maintenance to ECC 17.02.

## Task 1.3 Import Xpediter/*Xchange* License

When you received your Xpediter/*Xchange* product, you also received a license key for it. Import your Xpediter/*Xchange* license into the Compuware License Management System.



## Milestone 2: Install Xpediter/Xchange Using SMP/E

Xpediter/Xchange is installed using SMP/E. This milestone will guide you through the SMP/E installation of Xpediter/Xchange.



Roles involved:  
z/OS Security Administrator  
Xpediter/Xchange Installer.

### Task 2.1 Ensure Product Integrity



z/OS Security Administrator

Your site may already have a set of standards for SMP/E target and distribution libraries.

1. Provide the installation program with FULL access to the following libraries ([Table 7](#)).
2. Provide the individual responsible for product maintenance with UPDATE access.

**Table 7.** Access to SMP/E Libraries

Library	Identified By
Distribution	Low-level qualifier prefix ALXGxxxx
Target	Low-level qualifier prefix SLXGxxxx
Run-time	Run-time libraries and target libraries may be the same.



**Do not allow the installer to modify the distribution files.**

### Task 2.2 Follow the Compuware Installation Guide



Xpediter/Xchange Installer

1. Follow the instructions in the *Compuware Installer Mainframe Products SMP/E Installation Guide* to install Xpediter/Xchange.
2. Once completed, follow the steps in this guide to configure and deploy Xpediter/Xchange.



## Milestone 3: Specify Xpediter/Xchange Properties and Parameters

This milestone contains tasks that help prepare for configuration of Xpediter/Xchange for both a new install and an upgrade.



Roles involved:  
Xpediter/Xchange Installer

### Task 3.1 Update the MVS Program Properties Table

Make the following updates to the MVS Program Properties Table (PPT), SCHEDxx in SYS1.PARMLIB:

- Change the KEY parameter value to 4. This will allow Xchange to build its control blocks in CSA storage Key 4.
- Compuware recommends that you set the PRIV/NOPRIV parameter to PRIV and do **not** explicitly define Xchange in the WLM classification rules. The PRIV setting will allow Xchange to run in the SYSSVC service class. If Xchange is not allowed to execute at a high priority level, date/time exchanges may not occur.

```
PPT PGMNAME(XGMMAIN)
      CANCEL
      KEY(4)
      NOSYST
      PRIV
      NOSWAP
      DSI
      PASS
      AFF(NONE)
      NOPREF
```

Once the PPT Entry is made to the SCHEDxx member of SYS1.PARMLIB, use the MVS command SET SCH=XX to activate the new program properties table. XX represents the suffix appended to the SCHEDxx member in the SYS1.PARMLIB.

### Task 3.2 Implement the Compuware PARMLIB

Starting with release 17.02, Compuware mainframe products, including Xpediter/Xchange, use parameter libraries (PARMLIBs) in conjunction with the Compuware Mainframe Services Controller (CMSC) to configure each product as well as common components. The CMSC was installed as part of the Enterprise Common Components installation. For more information, see [PARMLIB Members](#) on page 14.

The default parameters that need to be specified in the Xchange CMSC PARMLIB member reside in member XCHG00 in the CPWR.MLXGmm.SLXGDATA library.

In this task, you will implement the Compuware PARMLIB for Xpediter/Xchange. This consists of creating a default member and updating the CMSC with it. The sample member is suffixed with 00 and should contain the values you want for the system on which Xpediter/Xchange is installed.

This task is necessary for both a new installation and an upgrade of Xpediter/Xchange to release 17.02.



Whenever you modify an existing Compuware PARMLIB member or add a new member, you will need to use the CMSC REFRESH command to update the contents of the CMSC cache.

### Task 3.2.1 Create the Default Member

Copy the associated member from the Xpediter/Xchange CPWR.MLXGnmn.SLXGDATA library members into your Compuware PARMLIB dataset. The SLXGDATA library members are set up for the most common settings.

### Task 3.2.2 Specify Xpediter/Xchange Parameters

The Xchange CMSC PARMLIB member name must be prefixed with XCHG and suffixed with either the CMSC default suffix or 1 to 4 characters of your choice. If suffixed with anything other than the default, make sure the suffix matches the SUFFIX= value on the PARM of the EXEC JCL statement in the Xchange startup JCL.



Commonly used and user-defined parameters are described in this Task. Descriptions of all other parameters are provided in the Xpediter/Xchange User Guide.

#### PARMLIB Syntax Rules

When editing an Xpediter/Xchange PARMLIB member, the following guidelines must be observed:

- All data is case sensitive.
- Each entry is to be coded as a *KEYWORD=value* format.
- Any characters in columns 73 through 80 are ignored.
- The Data-Area containing the *KEYWORD=value* information is columns 1 through 71, unless it is extended by a continuation.
- Placing a + character in column 72 denotes a continuation. The Data-Area is extended by 71 bytes by appending columns 1 through 71 of the following record to the end of the Data-Area.
- An asterisk (\*) in column 1 denotes a comment record:
  - A comment record cannot be continued.
  - All data on a comment record is ignored.
- Keywords are all upper case.
- The equality sign (=) must immediately follow the keyword.
- The value associated with the keyword begins immediately following the equality sign (=).
- Value termination rules:
  - If the first character of the value is a single-quote ('), the string is terminated by the last single-quote in the Data-Area.
  - Otherwise, if the first character of the value is a double-quote ("), the string is terminated by the last double-quote in the Data-Area.
  - Otherwise, if a space exists in the Data-Area, the value is terminated with the character immediately preceding the first space in the Data-Area.



- Otherwise, the value is terminated with the last byte of the Data-Area.
- Data following the last byte of the value in the Data-Area is ignored.

### Saving the CMSC PARMLIB Member and Refreshing

After adding the default parameters and modifying those parameters based on your organizational needs, save the *Xchange* CMSC PARMLIB member. Don't forget to enter the CMSC REFRESH command (see [Implement the Compuware PARMLIB](#) on page 23). The CMSC REFRESH command must be issued every time changes are made to the *Xchange* CMSC PARMLIB member. In addition, the *Xchange* subsystem must be recycled for the changes to take effect.

## Xchange Parameters

Essential *Xchange* parameter keywords, values, and defaults are defined below. Other parameters are described in the *Xpediter/Xchange User Guide*.

### SUBSYS

A four-character value that defines the subsystem ID for the *Xchange* program using this configuration. This value must be unique within the entire MVS complex. The subsystem ID used for a previous *Xchange* installation cannot be reused without performing an IPL. Compuware recommends consulting an MVS system programmer to choose a name that does not duplicate any existing MVS subsystem names. The default is CWXG.



*Xchange* dynamically creates this subsystem in the SSCVT. Do not add the subsystem to IEFSSNxx. No modification of SYS1.PARMLIB is required for installation, except to authorize the AUTHLIB load library.

The first *Xchange* subsystem started creates another subsystem named XGPC that is shared by any subsequent *Xchange* subsystems. The name XGPC is reserved for this *Xchange* subsystem and should not be used for any other subsystem.

### DMPPFX

A one- to twelve-character prefix used on all dump datasets that *Xchange* dynamically allocates. The default is CWXGDUMP.



The prefix you specify must conform to the rules for coding a dataset name. Do not end the prefix with a period (.). The period will be added when the dataset is created. If more than eight characters are used, a period (.) must be inserted to create two levels of the dataset qualifier. The 12-character maximum includes the period.

You must ensure that your external security system gives all userIDs authority to create new dump datasets that are named using the prefix you specify.

### SECPFX

A one- to eight-character name that the external security module uses as a prefix when creating entity names to pass to the security system. The default is CWXG. For information on external security see [Milestone 8: Configure Xchange External Security](#).

## DEFINE\_JOURNALS and END Block Parameters



If you choose to have a journal or journal datasets, DEFINE\_JOURNALS, the block start parameter, must immediately precede the first related block LOGxxx parameter, and the END block end parameter must immediately follow the last one.

Use the LOGxxx parameters to define each journal dataset used by the *Xchange* subsystem. There is no limit to the number of journal datasets that a single *Xchange* subsystem can use. You can define as few or as many as you like, but you must specify one LOGDDN, LOGDSN, and LOGCOPY parameter for each journal dataset.

Each journal dataset must have been defined by IDCAMS, and there cannot be JCL statements in the *Xchange* job referring to these datasets. *Xchange* dynamically allocates each dataset and, in turn, de-allocates each dataset when it is filled. This process allows automatic or manual unloading of data from inactive journal datasets.

The JCL statements used to create your journal datasets reside in the installation library CPWR.MLXGmm.SLXGCNTL file member ALLOCLOG. Make a note of the journal file names you use in this macro so you can enter them accurately in the ALLOCLOG JCL deck in [Allocate the Journal Datasets](#) on page 31.

If you intend to view or extract records from the journal, specify at least two journal files. This permits automatic swapping from a filled journal file to the next journal file. When each journal file is swapped out, the next file is reset when it is swapped in, and data in it is lost. Therefore, more journal files give you more flexibility in monitoring *Xchange* transactions, but they require space allocated for the files themselves.



Discuss the intended usage of *Xchange* with your site's users to determine optimal values for the LOGxxx parameters.

### DEFINE\_JOURNALS

Block start parameter to identify the beginning of related parameters that define each journal dataset used by the *Xchange* subsystem.

This parameter must precede any LOGxxx parameters.

#### LOGDDN

The one- to eight-character name of the data definition (DD) statement that *Xchange* uses when referencing this journal dataset.

#### LOGDSN

The 1- to 44-character dataset (DS) name for this journal dataset.

#### LOGCOPY

Whether the journal dataset is the backup journal file. Each time a journal file is filled, all date records are copied from that file to the backup journal file and appended to the existing entries in that backup file. YES indicates that the journal file is the backup file. NO indicates that it is not. Only one LOGCOPY parameter can contain the YES value. The default is NO.

### END

Block end statement for DEFINE\_JOURNALS parameter block.

This parameter must immediately follow any LOGxxxx parameters.

The parameters prefixed with “LOG” above must be in the order listed for each journal dataset. For example, if using multiple journal datasets, the parameters would be listed in the following order:



```
DEFINE_JOURNALS
  LOGDDN=JOURNAL1
  LOGDSN=CUSTOMER.JOURNAL1
  LOGCOPY=NO
  LOGDDN=JOURNAL2
  LOGDSN=CUSTOMER.JOURNAL2
  LOGCOPY=NO
  etc.
END
```

## PREVDATE

Whether previous dates are allowed (that is, whether a date can be exchanged for a date previous to the current date). The default is NO.

## REPLY

Whether an outstanding reply message will display on the operator console. The default is NO.



The console operator can always enter *Xchange* commands with the MVS MODIFY and STOP commands.

## RESTREQ

The restore requests parameter used to control the *Xchange* save and restore process during startup and normal processing of the *Xchange* address space. Valid values are WARM, COLD, and OFF.

Specifying WARM will cause *Xchange* to restore requests saved from its previous run. All requests are restored unless *Xchange* configuration changes, such as a reduction in XREs, prevent the restore. Jobclass, COPE Logical System, and CICS H requests are restored with the same first-use running date/time as when they were saved. The default is WARM.

Specifying COLD will cause *Xchange* to bypass restore processing during startup and delete any requests in the save/restore file. This option can be used to bypass restore processing if the file becomes damaged or to delete all current *Xchange* requests.

Requests created after *Xchange* is started with RESTREQ=COLD are still saved, however, and can be used at the next *Xchange* startup if the value of RESTREQ is subsequently changed to WARM.

Setting this RESTREQ to OFF will cause *Xchange* to bypass all save and restore processing. If your site uses the OFF setting, the XGSAV001 DD statement can be deleted from the XGSTART JCL or *Xchange* startup process.

## Task 3.2.3 Update the CMSC with your PARMLIB Members

Use the z/OS MODIFY (F) command to update the CMSC with the PARMLIB members you created.

### Refreshing All PARMLIB Members

```
F cmscname,PARMLIB REFRESH
```

## Refreshing a Single Parameter Member

```
F cmscname,PARMLIB REFRESH member_name
```



During the refresh process, parameter values are validated. If they are found to be invalid, an error message is written to the FDBDLOG SYSOUT dataset associated with the ECC CMSC instance where the member is being refreshed. If an error is detected, the contents of the in-core member will not be refreshed. You must correct the error and refresh the member before attempting to use it.

## Milestone 4: Update TSO CLISTs

This milestone contains tasks for copying and updating Xpediter/*Xchange* TSO CLISTs for both a new install and an upgrade.



Roles involved:  
Xpediter/*Xchange* Installer

### Task 4.1 Copy Members

Copy SLXGCLIB members HELPPTFS, XCHANGE, XGCGDSN, XGFNDIT, and XGLOG into a library in your SYSPROC concatenation.

This CLIST library **must** be allocated to your ISPF session.



You can display the libraries in your SYSPROC concatenation by entering the command  
TSO ISRDDN.

### Task 4.2 Edit the CLISTs

By default, each CLIST (except XGCGDSN) will use the Simple Deploy method to allocate the dataset names specified in the CSMC PARMLIB DDSN\* member (DDSN00 by default). If you are not using the default member, make sure to point to the DDSN\* member name by setting the DDSNSFX keyword to the one- to four-character PARMLIB member suffix.

For example, to point to PARMLIB member DDSNCPWR, use:

```
DDSNSFX(CPWR)
```

In addition, the primary CMSC will be used by default. If you are not using the default CMSC, specify the CMSC to use by setting the CMSC keyword to the one- to four-character CMSC ID.

For example, to point to CMSC subsystem TEST, use:

```
CMSC(TEST)
```

This CLIST can alternately use High-Level Qualifiers (HLQ) to allocate the Xpediter/*Xchange* datasets it references. See the comments in each CLIST for further information.



XGCGDSN is an enablement CLIST and requires no modification.



# Milestone 5: Allocate Journal and Save/Restore Datasets

This milestone contains tasks for allocating the Xpediter/*Xchange* journal datasets and save/restore datasets for both a new install and an upgrade.



Roles involved:  
Xpediter/*Xchange* Installer

## Task 5.1 Allocate the Journal Datasets

1. Use IDCAMS to create each journal dataset.

You do not need to pre-initialize the datasets. They simply need to be created. [Figure 1](#) provides a sample of an IDCAMS job for creating journal datasets. This JCL is included in the CPWR.MLXGmm.SLXGCNTL PDS as member ALLOCLOG.

2. Edit ALLOCLOG following the instructions given near the top of the file, then submit the job.

**Figure 1.** Sample IDCAMS job for Allocating a Journal Dataset (ALLOCCLOG)

```

/** --> PLACE JOB CARD HERE
/**
/*******
/** INSTALLER: *
/** *
/** TO USE THIS JCL, YOU NEED TO *
/** *
/** - USE ISPF 'CHANGE ALL' TO REPLACE 'CUSTOMER.JOURNALN' (WHERE *
/** N=A DIGIT CORRESPONDING TO THOSE IN YOUR JOURNAL DSNS) WITH *
/** THE JOURNAL FILE DSNS YOU WISH TO USE WITH XPEDITER *
/** XCHANGE. NOTE: THESE MUST BE THE SAME NAMES AS THOSE YOU *
/** USED WHEN YOU EDITED THE XCHANGE PARMLIB MEMBER IN YOUR *
/** SITE'S COMMON PARMLIB DATASET FOR COMPUWARE. IF YOU ARE *
/** NOT CERTAIN OF THOSE NAMES, REFER TO THE XCHANGE PARMLIB *
/** MEMBER IN YOUR SITE'S PARMLIB DATASET FOR COMPUWARE. *
/** *
/** - USE ISPF 'CHANGE ALL' TO REPLACE 'VOLSER' WITH THE VOLUME *
/** SERIAL NUMBER OF A DASD UNIT ON WHICH YOU CAN CREATE VSAM *
/** FILES. *
/** *
/** - INSERT A VALID JOB CARD AND SUBMIT THE JOB. *
/*******
/**
//DELETE EXEC PGM=IDCAMS,REGION=3M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE customer.journal1 CLUSTER
DELETE customer.journal2 CLUSTER
DELETE customer.journal3 CLUSTER
/*
//DEFINE EXEC PGM=IDCAMS,REGION=3M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DEFINE CLUSTER (NAME(customer.journal1) -
TRK(90) VOL(volser) -
CISZ(16384) SPANNED -
RECSZ(60 32600) NONINDEXED -
REUSE SHR(2 3) RECOVERY)
DEFINE CLUSTER (NAME(customer.journal2) -
TRK(90) VOL(volser) -
CISZ(16384) SPANNED -
RECSZ(60 32600) NONINDEXED -
REUSE SHR(2 3) RECOVERY)
DEFINE CLUSTER (NAME(customer.journal3) -
TRK(90) VOL(volser) -
CISZ(16384) SPANNED -
RECSZ(60 32600) NONINDEXED -
REUSE SHR(2 3) RECOVERY)
/*

```

## Task 5.2 Allocate the Save/Restore Dataset

*Xchange* can save all date/time simulation requests to a save/restore dataset at user-specified intervals, then automatically restore them when *Xchange* is restarted.



The save/restore dataset is created with IDCAMS using the sample JCL shown in [Figure 2](#). This JCL is provided in member CPWR.MLXGmm.SLXGCNTL(ALLOCSAV). Edit the ALLOCSAV member following the instructions given near the top of the file, then submit the job.



If your site does not want to use *Xchange's* save and restore capabilities:

- Do not submit the JCL shown in this step.
- Specify OFF for the RESTREQ *Xchange* parameter. See [RESTREQ](#) on page 27 for more information.

**Figure 2.** SAVE/RESTORE Dataset Allocation JCL

```

/** ==> INSERT JOB CARD HERE
/**
/*******
/** INSTALLER: *
/** *
/** TO USE THIS JCL, YOU NEED TO *
/** *
/** - USE ISPF 'CHANGE ALL' TO REPLACE 'customer.saverest.file' *
/** WITH THE FILE DSN YOU WISH TO USE WITH XPEDITER/XCHANGE. *
/** *
/** NOTE: THIS MUST BE THE SAME NAME AS THE NAME SUPPLIED IN *
/** THE XCHANGE START JCL FOR THE DD XGSAV001. *
/** *
/** - USE ISPF 'CHANGE ALL' TO REPLACE 'volser' WITH THE VOLUME *
/** SERIAL NUMBER OF A DASD UNIT ON WHICH YOU CAN CREATE VSAM *
/** FILES. *
/** *
/** - INSERT A VALID JOB CARD AND SUBMIT THE JOB. *
/** *
/*******
/**
/**
//DELETE EXEC PGM=IDCAMS,REGION=3M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE customer.saverest.file CLUSTER
/*
//DEFINE EXEC PGM=IDCAMS,REGION=3M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DEFINE CLUSTER (NAME(customer.saverest.file) -
CYL(1) VOL(volser) -
CISZ(32768) -
NONINDEXED -
REUSE -
RECOVERY -
RECSZ(100 32758) -
SHR(2 3))

```



# Milestone 6: Enable Language Environment (LE) Support

This milestone contains tasks for enabling Language Environment (LE) in Xpediter/*Xchange* for CICS region support, DB2 support, IMS message level support, PL/I support, and C support.



Roles involved:  
Xpediter/*Xchange* Installer

## Task 6.1 Enable LE for CICS Region Support



If your site does not require date/time exchanges for LE programs under *Xchange* CICS region support, you may skip this task and continue with [Enable LE for DB2 Support](#).

### Language Environment/CICS Dependency Information

Xpediter/*Xchange* supports Language Environment (LE). Ensure that your site is current on z/OS, CICS, and LE maintenance before attempting to use Language Environment with *Xchange*.

#### Task 6.1.1 Apply CEEPLPKA Zap

A zap to LE module CEEPLPKA is required if you want to exchange LE in CICS at the region level. The zap is necessary because LE gets initialized in a way that circumvents the normal dynamic hooking of LE by *Xchange*. To enable this support, perform the following steps:

1. Compuware recommends that a copy of load module CEEPLPKA be placed in its own library and this zap be applied to that copy.
2. Run the job contained in dataset CPWR.MLXGnnn.SLXGCNTL member XGZLE370. Follow the directions contained in the comments of the job.



For customers running z/OS 2.3 and above, the zap to LE module CEEPLPKA is located in dataset CPWR.MLXGnnn.SLXGDATA member XGZPLE23. This zap works in conjunction with *Xchange's* Enhanced STCKE support. For more information, see [Milestone 7: Enable Xchange SVCs](#). The zap must be edited to specify the assigned user SVC number to be used by *Xchange's* Enhanced STCKE support.

- Place the library with the zapped version of CEEPLPKA in the DFHRPL DD of the CICS region that is to be exchanged. This library must be concatenated before the normal LE SCEERUN library. Only use this zapped copy of CEEPLPKA when *Xchange* is required in a CICS region.

Whenever maintenance is applied to LE, make a new copy of CEEPLPKA and reapply this zap to that new copy.



If region level CICS exchanges are required and this zap has been applied, *Xchange* CICS transaction level support should **not** be used. Mixing region level and transaction level support concurrently in a CICS environment may cause the date/time value in the EIB to differ from the date and time requested.

If you will also be performing the procedure in [Enable LE for DB2 Support](#) on page 36, you can use the same zapped CEEPLPKA module from this step to exchange DB2 stored procedures.

## Task 6.2 Enable LE for DB2 Support



If your site does not require date/time exchanges for LE programs under *Xchange* DB2 stored procedure support, you may skip this task and continue with [Enable Enterprise PL/I DATE or DATETIME Built-in Function Support](#).

A zap to LE module CEEPLPKA is required if you want to exchange date/time calls from within DB2 stored procedures. The zap is necessary because LE gets initialized in a way that circumvents the normal dynamic hooking of LE by *Xchange*.



If you already executed CPWR.MLXGnnn.SLXGCNTL member XGZLE370 to create a zapped version of CEEPLPKA in [Enable LE for CICS Region Support](#) on page 35, you can use the same zapped module to exchange DB2 stored procedures.

To enable LE for DB2 support, perform the following steps:

- Compuware recommends that a copy of load module CEEPLPKA be placed in its own STEPLIB/JOBLIB and this zap be applied to that copy.
- Run the job contained in dataset CPWR.MLXGnnn.SLXGCNTL member XGZLE370. Follow the directions contained in the comments of that job. This STEPLIB/JOBLIB must be concatenated before the normal LE SCEERUN library in both the DB2 WLM JCL and the JCL invoking the DB2 application. Only use this zapped copy of CEEPLPKA when *Xchange* is required from within the execution of the DB2 stored procedure.



For customers running z/OS 2.3 and above, the zap to LE module CEEPLPKA is located in dataset CPWR.MLXGnnn.SLXGDATA member XGZPLE23. This zap works in conjunction with *Xchange's* Enhanced STCKE support. For more information, see [Milestone 7: Enable Xchange SVCs](#). The zap must be edited to specify the assigned user SVC number to be used by *Xchange's* Enhanced STCKE support.

- Whenever maintenance is applied to LE, make a new copy of CEEPLPKA and reapply this zap to that new copy.



If Xpediter/*Xchange* will be working with LE date/time calls in an IMS message processing region (MPR) that specifies a pre-load of CEELRRIN, you must have the *Xchange* zap to CEEPLPKA in the STEPLIB concatenation *before* the SCEERUN library. If you created a zapped CEEPLPKA module in [Enable LE for CICS Region Support](#) on page 35, you can use that same zapped CEEPLPKA module in this step. An alternative solution is to remove CEELRRIN from the pre-load list.

## Task 6.3 Enable Enterprise PL/I DATE or DATETIME Built-in Function Support

*Xchange* supports both the DATE and DATETIME built-in functions of Enterprise PL/I.



- If your site does not require date/time exchanges for the Enterprise PL/I DATE or DATETIME built-in functions, you may skip this step and continue with [Enable C time\(\) and gettimeofday64\(\) Function Support](#).
- If your site is using the Enterprise PL/I DATE or DATETIME built-in function without any date/time patterns, you may skip this step and continue with [Enable C time\(\) and gettimeofday64\(\) Function Support](#). An example *without* a date/time pattern is **DATETIME()**. An example *with* a date/time pattern is **DATETIME(YYYYMMDD)**.

A zap to LE module CEEEV003 is required if you want to exchange the Enterprise PL/I DATE or DATETIME built-in functions while using date/time patterns. To enable this support, perform the following steps:

1. Compuware recommends that a copy of the load module CEEEV003 be placed in its own library and this zap be applied to that copy. CEEEV003 can be found in the LE SCEERUN library.
2. This zap works in conjunction with *Xchange's* Enhanced STCK support or *Xchange's* Enhanced STCK and Enhanced STCKE support. See [Milestone 7: Enable Xchange SVCs](#).
3. Run the job contained in dataset CPWR.MLXGnnn.SLXGCNTL member XGZENTPL. Follow the directions contained in the comments of the job. Editing of CPWR.MLXGnnn.SLXGDATA member XGZAPPLI or XGZAPP23 is required in order to supply the user SVC number(s) to be used by *Xchange's* Enhanced STCK or *Xchange's* Enhanced STCK and Enhanced STCKE Support, respectively.
4. Concatenate the library with the zapped version of CEEEV003 before the normal LE SCEERUN library. Only use this zapped copy of CEEEV003 when *Xchange* is required to support the PL/I DATE or DATETIME built-in function.



Whenever maintenance is applied to LE, make a copy of CEEEV003 and reapply this zap to that new copy.

## Task 6.4 Enable C time() and gettimeofday64() Function Support



If your site does not require date/time exchanges for the C time() or gettimeofday64() function, you may skip this step and continue with [Milestone 7: Enable Xchange SVCs](#).

A zap to LE module CEEEV003 is required if you want to exchange the C time() and gettimeofday64() functions.



If you already executed CPWR.MLXGnnn.SLXGCNTL member XGZENTPL to create a zapped version of CEEEV003 in [Enable Enterprise PL/I DATE or DATETIME Built-in Function Support](#) on page 37, you can use the same zapped module to exchange C time() and gettimeofday64() functions.

To enable this support, perform the following steps:

1. Compuware recommends that a copy of the load module CEEEV003 be placed in its own library and this zap be applied to that copy. CEEEV003 can be found in the LE SCEERUN library.
2. This zap works in conjunction with *Xchange's* Enhanced STCK support or *Xchange's* Enhanced STCK and Enhanced STCKE support. See [Milestone 7: Enable Xchange SVCs](#).

3. Run the job contained in dataset CPWR.MLXGmm.SLXGCNTL member XGZENTPL. Follow the directions contained in the comments of the job. Editing of the CPWR.MLXGmm.SLXGDATA member XGZAPPLI or XGZAPP23 is required in order to supply the user SVC number(s) to be used by *Xchange's* Enhanced STCK or *Xchange's* Enhanced STCK and Enhanced STCKE Support, respectively.
4. Concatenate the library with the zapped version of CEEEV003 before the normal LE SCEERUN library. Only use this zapped copy of CEEEV003 when *Xchange* is required to support the C time() or gettimeofday64() function.



Whenever maintenance is applied to LE, make a copy of CEEEV003 and reapply this zap to that new copy.

## Milestone 7: Enable *Xchange* SVCs

This milestone contains tasks for enabling an *Xpediter/Xchange* SVC for Enhanced STCK Support and STCKSYNC macro support *and* another *Xpediter/Xchange* SVC for Enhanced STCKE Support and STCKE macro support. These are two separate and distinct SVCs. They are described in more detail in the tasks below.



Roles involved:  
Xpediter/*Xchange* Installer  
z/OS System Programmer

### Starting *Xchange* as a Started Task

Starting *Xchange* as a started task automatically executes the product at IPL and uses no initiators. Also, started tasks are not constrained by batch performance values.

To execute *Xchange* from the operator console as a started task, you will place the procedure shown in [Figure 3](#) in the appropriate procedure library (PROCLIB). Be sure to include the TIME=1440 parameter shown to prevent the job from exceeding the CPU time limit (MVS abend S322).

**Figure 3.** Sample Command Procedure to Initiate *Xchange*

```
//XCHANGE PROC
//IEFPROC EXEC PGM=XGMMAIN,TIME=1440,
// PARM='SVCNUM=???,RUNTYPE=INSTALL,SVCENM=???,RUNETYP=INSTALL'
//STEPLIB DD DSN=CPWR.MLXGmm.SLXGAUTH,DISP=SHR
//XGSAV001 DD DSN=customer.saverest.file,DISP=SHR
```

In the example shown in [Figure 3](#), you must provide the following information:

- Your APF authorized library in place of CPWR.MLXGmm.SLXGAUTH. For more information, see [Designate an APF Authorized Library](#) on page 13.
- Your save/restore dataset in place of customer.saverest.file. For more information, see [Allocate the Save/Restore Dataset](#) on page 32.

## Tasks

Complete the following tasks to enable *Xchange* SVC support.

### Task 7.1 Enable SVC for Enhanced STCK and STCKSYNC Macro Support

*Xchange* provides support for Enhanced STCK Support for date/time simulation of Assembler STCK operation code and STCKSYNC macro support. STCK and STCKSYNC support are implemented with one SVC. The SVC is installed as part of the *Xchange* startup JCL.

1. Consult your site's system programmer for an available SVC number between 200 and 255.  
Ensure the SVC number chosen will not be used by any other product.

2. Edit the JCL shown in [Figure 3](#), replacing the ??? values on the field SVCNUM with the SVC number.
3. Specify INSTALL or REFRESH for RUNTYPE.

Use INSTALL unless specifically told to use REFRESH by *Xchange* Customer Solutions. When INSTALL is used, a check is done to see if the SVC is installed. If it is not installed, *Xchange* will dynamically install the specified SVC. The SVC will remain installed until the next system IPL. Between IPLs, using INSTALL during the startup of *Xchange* ensures that the SVC is installed.

4. When you are ready to start *Xchange*, submit the revised JCL.

## Using STCKSYNC Support

STCKSYNC support is required with CICS region support. The activation of STCKSYNC support is done during the startup of *Xchange*. By enabling the *Xchange* SVC for Enhanced STCK Support, you also enable STCKSYNC support. See [Enable CICS Region Support](#) on page 54 for additional information.

## Task 7.2 Enable SVC for Enhanced STCKE and STCKE Macro Support

*Xchange* provides support for Enhanced STCKE Support for date/time simulation of Assembler STCKE operation code and STCKE macro support. The SVC is installed as part of the *Xchange* startup JCL.

1. Consult your site's system programmer for an available SVC number between 200 and 255.  
Ensure the SVC number chosen will not be used by any other product.
2. Edit the JCL shown in [Figure 3](#) on page 39, replacing the ??? values on the SVCENM field with the SVC number.
3. Specify INSTALL or REFRESH for RUNETYP.

Use INSTALL unless specifically told to use REFRESH by *Xchange* Customer Solutions. When INSTALL is used, a check is done to see if the SVC is installed. If it is not installed, *Xchange* will dynamically install the specified SVC. The SVC will remain installed until the next system IPL. Between IPLs, using INSTALL during the startup of *Xchange* ensures that the SVC is installed.

4. When you are ready to start *Xchange*, submit the revised JCL.



If *Xchange* is started without installing the SVC, any zapped modules using Enhanced STCK or STCKE Support will fail with an Sfix abend, where xx is the hex representation of the SVC number used in the zap.

For information on how to use Enhanced STCK or STCKE Support, see the *Xpediter/Xchange User Guide*.

## Using Enhanced STCKE Support

Enhanced STCKE support is required for systems running z/OS 2.3 and above that also need to have the zap to LE module CEEPLPKA applied for any of the following reasons:

- CICS region support,
- DB2 stored procedure support, or
- If *Xpediter/Xchange* will be working with date/time calls in an IMS MPR that specifies a pre-load of CEELRRIN.



# Milestone 8: Configure *Xchange* External Security

This milestone contains tasks for configuring external security. If you are using an external security package (such as RACF, CA-ACF2®, or CA-TOP SECRET®) at your installation, you will need to configure *Xchange* with the package.



Roles involved:  
Xpediter/*Xchange* Installer  
Security Administrator

## Overview

*Xchange* makes external security calls to protect jobs, steps, and programs from unauthorized execution using date and time modification. External security is invoked by *Xchange* when the job designated in a pattern job request executes an SVC11 or other TIME services call — not when the request is created.

You can define a default action that is taken whenever a job selected by *Xchange* is not covered by a security access rule. You can specify that all such jobs are allowed either to run or to fail.

When you create the *Xchange* CMSC PARMLIB member, the following security options are available:

- **SECUSE=NO** does not activate external security checking. If you specify this parameter, the **SECDFLT** parameter is ignored.
- **SECUSE=YES** activates external security checking. The following choices are then available for the **SECDFLT** parameter:
  - **SECDFLT=ALLOW** sets the security default to allow jobs that are not covered by a security rule to have their dates and times changed by *Xchange*.
  - **SECDFLT=DENY** sets the security default to fail date and time change requests that are not covered by a security rule.

## Task 8.1 Assemble and Link the RACROUTE Module



Xpediter/*Xchange* Installer  
Security Administrator

The RACROUTE module found in install CPWR.MLXG*mm*.SLXGDATA file member USRRACRT is designed to work with any SAF-compliant external security package. This module may be modified, if required.

JCL to assemble and link the RACROUTE module (USRRACRT) is included in the CPWR.MLXG*mm*.SLXGCNTL PDS as member ASMRACRT. Make any necessary edits and run the job.

## Task 8.2 CA-TOP SECRET and ACF2 Configuration



### Security Administrator

1. Sites using CA-TOP SECRET should check their facility definitions and ensure that the RES/NORES option is set to RES. If this option is set to NORES, DB2 and IMS date and time simulations will not occur properly.
2. Sites using ACF2 must set up SAFDEF records similar to the following:

```
SAFDEF.XCHANGE
      FUNCRET(0)  FUNCRSN(0)  ID(XCHANGE)  MODE(GLOBAL)  PROGRAM(-)
      RACROUTE(REQUEST=AUTH CLASS=DATASET) RB(-) RETCODE(4)
```

3. If your site uses the ACF2 Command List Table, define the following TSO commands:
  - XGIPGM01
  - XGIZAPL
  - XGIZAPA

Consult your ACF2 administrator.

## Task 8.3 Define Security Rules



### Xpediter/Xchange Installer Security Administrator

The SECPFX parameter is used to define a one- to eight-character prefix for security pseudo-dataset names (see [SECPFX](#) on page 25). The default of CWXG can be changed to suit site requirements. Whatever prefix is chosen should be unique in that it does not duplicate any existing dataset name prefix within the installation's system.



If RACF is the security access method you are using, this prefix must be defined as a group or user ID to RACF.

A pseudo-dataset name is built for each *Xchange* request. Its format is

```
prefix.jobname.stepname.procstep.pgmname
```

The values for jobname, stepname, procstep, and pgmname are taken from the *Xchange* request specified for the step and from the system at the time the job makes an SVC11 or other TIME services call.



If a job is being run as a started task and has no procstep, the pseudo-dataset name would have the format `prefix.jobname.stepname.pgmname`

To prevent a TSO userID from being exchanged, deny update access to the pseudo-dataset name `prefix.userID.**`.

### Task 8.3.1 Establish User ID Access

The user ID of a pattern job request's owner must have update access to the pseudo-dataset name generated by *Xchange*. Otherwise they will be prevented by external security from using *Xchange* to make a date and time change.

The following scenario provides a basic overview of how *Xchange* would work with a typical site's external security package:

1. An *Xchange* user creates a pattern job request.
2. A job step included in the request makes an SVC11 or other TIME services call.
3. *Xchange* builds a pseudo-dataset name as described above.
4. *Xchange* invokes the external security package to verify that the request owner's user ID has authority for update access to the pseudo-dataset name.
5. If the request owner's user ID is authorized for update access to the pseudo-dataset name, *Xchange* makes the date and time change. Otherwise, the date and time for the step are not exchanged.

The pseudo-dataset name must be defined as CLASS=DATASET and, as such, must not duplicate any real dataset names defined to the security system. Duplication is avoided by using the SECPFX parameter to specify a unique pseudo-dataset name prefix.

The user ID under which the target job runs plays no part in the *Xchange* security checking. For that reason, it does not matter who submits the target job. It can be submitted by any production or testing mechanism, including a job scheduling package.

If the *Xchange* batch facility is used to make the request, the user who submits the job will always be the same one who made the request. That user must have update access to the *Xchange* pseudo-dataset name.

### Task 8.3.2 Define Security Rules for Jobclass Support

If *Xchange* is configured to use external security (SECUSE=YES, SECDFLT=DENY), you must define additional profiles/rules to your security system in order to be able to use the line commands on the Simulated Date Time Settings screen in jobclass format.

As explained above, the SECPFX parameter is used to define a one- to eight-character prefix for security pseudo-dataset names (see [SECPFX](#) on page 25). The default of CWXG can be changed to suit site requirements. Whatever prefix is chosen should be unique in that it does not duplicate any existing dataset name prefix within the installation's system.



If RACF is the security access method you are using, this prefix must be defined as a group or user ID to RACF.

A pseudo-dataset name is built for each *Xchange* jobclass request. Its format is:

```
prefix.JOBCLASS.#x
```

or

```
prefix.JOBCLASS.#class1.#class2
```

The **x** in **#x** represents a valid 1- to 7-character job class. Valid default job classes are A to Z and 0 to 9. The **class1** in **#class1** represents the first 7 characters of an 8-character job class, and **class2** in **#class2** represents the last character of an 8-character job class. Your *Xchange* administrator—or whoever will be responsible for enabling, disabling, and resetting jobclass *Xchange* requests—must have update access to the pseudo-dataset name generated by *Xchange*. If they do not, a NAUT (not authorized) error message will be displayed in the Imsg field of the Simulated Date Time Settings screen in jobclass format when a line command is entered for the corresponding jobclass.

For example, with SECPFX=CWXG the following profiles/rules could be defined to your external security package:

```
CWXG.JOBCLASS.** All jobclasses would be eligible for Xchange jobclass requests.
```

```
CWXG.JOBCLASS.#A Jobclass A would be eligible for Xchange jobclass requests.
CWXG.JOBCLASS.#P Jobclass P would be eligible for Xchange jobclass requests.
CWXG.JOBCLASS.#0 Jobclass 0 would be eligible for Xchange jobclass requests.
```

Once a JOBCLASS is activated with the H line command, any user can submit jobs in that jobclass for date and time exchange. There are no other security checks made by *Xchange* for jobs being exchanged under jobclass support.

### Task 8.3.3 Define Security Rules for DB2 Distributed Data Facility (DDF) Support

If *Xchange* is configured to use external security (SECUSE=YES, SECDFLT=DENY), and the DDF parameter is set to YES, you must define additional profiles/rules to your security system in order to exchange entries that are added by the DDF format of the Simulated Date Time Settings screen.

As explained above, the SECPFX parameter is used to define a one- to eight-character prefix for security pseudo-dataset names. (For more information, see [SECPFX](#) on page 25). The default of CWXG can be changed to meet site requirements. Whatever prefix is chosen should be unique in that it does not duplicate any existing dataset name prefix in the installation's system.



If RACF is the security access method you are using, this prefix must be defined as a group or user ID to RACF.

A pseudo-dataset name is built for each *Xchange* DDF request. *Xchange* will use a pseudo-dataset name constructed according to one of the following two rules in which *ccbresnm* is the abbreviation for the resource name field value and *db2ssid* is the abbreviation for the DB2 SSID field value.

- If the correlation ID is comprised of a valid jobname identifier followed by spaces, the pseudo-dataset name format is:

```
prefix.jobname.#.ccbresnm.db2ssid
```

- Otherwise, the correlation ID's contents is translated and split as follows:
  - Numbers, uppercase letters, #, \$, and @ remain unchanged
  - Lowercase letters are translated to uppercase
  - All remaining characters are translated to a #
  - The resultant twelve-byte field is split into two six-byte fields, *ccbcor1* and *ccbcor2*.

The pseudo-dataset name format is:

```
prefix.#ccbcor1.#ccbcor2.ccbresnm.db2ssid
```

#### Example 1

Assume the SECPFX parameter is set to CWXG. The correlation ID and resource name of the job making the DB2 SQL DDF request are TSOUSR1A and XGBCHDB2, respectively. Assume the *Xchange* DDF request is being restricted to DB2 Subsystem D14G.

Because TSOUSR1A satisfies the rules for being a valid jobname, the first rule above is used to generate the following pseudo-dataset name:

```
CWXG.TSOUSR1A.#.XGBCHDB2.D14G
```

#### Example 2

Again assume the SECPFX parameter is set to CWXG. The correlation ID and resource name of the job making the DB2 SQL DDF request are javaw.exe and DISTSERV, respectively.

Because javaw.exe does not satisfy the rules for being a valid jobname, the second rule above is used to generate the following pseudo-dataset name:

CWXG.#JAVAW#.#EXE###.DISTSERV

In this example, javaw.exe becomes JAVAW#EXE### because the period and spaces become #s, and the lowercase letters are uppercased. It is then split into two six-byte fields, JAVAW# and EXE###.

### Task 8.3.4 Establish Resource Rules

Generic resource rules apply to the *Xchange* security pseudo-dataset names. For complete protection, an installation could specify SECDFLT=DENY, define each dataset resource, and allow only specific user IDs to have update access to those resources. Another alternative would be to define a security rule for a “catch all” pseudo-dataset name that would apply if no other more restrictive pseudo-dataset name rule could be applied. This pseudo-dataset name would be specified as CWXG.\*\*, where CWXG is the value of the SECPFX parameter. The security rule for this pseudo-dataset name would be defined with universal access (UACC) equal to NONE, and no user IDs or groups would have update access to it.

More restrictive security rules could be defined. For example, assume that an installation has a set of jobs for payroll processing, and job names for these jobs are PAYPROD (for production runs) and PAYTEST (for test runs). *Xchange* protects the installation’s PAYPROD jobs so that these jobs never have their dates and times modified, while allowing certain user IDs or groups the ability to modify the dates and times of PAYTEST jobs. The installation would define CWXG.PAYPROD.\*\* as UACC=NONE with no user IDs or groups having update access. CWXG.PAYTEST.\*\* would be defined as UACC=NONE with selected user IDs and groups having the required update access.

PROCSTEP and STEP names can also be reflected in the security pseudo-dataset name. For example, assume a production job named ACCTPROD has a step named UPDATE that executes a PROC. This PROC has a step named NAMAST. The pseudo-dataset name security rule to protect the step would be CWXG.ACCTPROD.UPDATE.NAMAST.\*\*.

Program names are specified last on the security rule’s pseudo-dataset name. To protect a program named UPDTPGM, specify a pseudo-dataset name of CWXG.\*\*.UPDTPGM.

With these security rules, complete protection of the installation’s jobs can exist, while still allowing authorized users to run jobs with dates and times modified.

### Task 8.3.5 Designate *Xchange* Administrators

Only *Xchange* administrators can perform the following functions:

- Delete pattern requests and jobs created by a different user ID
- Input operator commands from the batch facility
- Reset pattern requests on the ISPF interface created by a different user ID
- Delete all completed entries on the ISPF interface for jobs not active
- Force active pattern requests and their executing requests on the ISPF interface to become inactive/queued and completed.

*Xchange* administrators have user IDs with update access to the model dataset entity name XGADMIN. XGADMIN has been prefixed by the SECPFX parameter. For example, if SECPFX is specified as CWXG, and an installation wants user IDs USER1 and USER2 to be *Xchange* administrators, the installation would define a dataset model named CWXG.XGADMIN and would give update access to USER1 and USER2.

To give administrator functions to all user IDs, the installation would define a dataset model named CWXG.XGADMIN and would give it universal access equal to UPDATE.

To prohibit administrator functions from all user IDs, the installation would define a dataset model named CWXG.XGADMIN with universal access equal to NONE.



If external security is not activated (SECUSE=NO), all user IDs are considered *Xchange* administrators.

### Task 8.3.6 Define Security Rules for COPE Logical System Support

If *Xchange* is configured to use external security (SECUSE=YES, SECDFLT=DENY), you must define additional profiles/rules to your security system to be able to use the line commands on the Simulated Date Time Settings screen in COPE Logical System format.

As explained above, the SECPFX parameter is used to define a one- to eight-character prefix for security pseudo-dataset names (see [SECPFX](#) on page 25). The default of CWXG can be changed to suit site requirements. Whatever prefix is chosen should be unique in that it does not duplicate any existing dataset name prefix within the installation's system.



If RACF is the security access method you are using, this prefix must be defined as a group or user ID to RACF.

A pseudo-dataset name is built for each *Xchange* COPE Logical System request. Its format is:

*prefix*.COPELSYS.*lsys*

where *lsys* represents a valid 1- to 8-character COPE Logical System name. Your *Xchange* administrator—or whoever will be responsible for enabling, disabling, and resetting COPE Logical System *Xchange* requests—must have update access to the pseudo-dataset name generated by *Xchange*. If they do not, a NAUT (not authorized) error message will be displayed in the Imsg field of the Simulated Date Time Settings screen in COPE Logical System format when a line command is entered for the corresponding COPE Logical System.

For example, with SECPFX=CWXG the following profiles/rules could be defined to your external security package:

- CWXG.COPELSYS.\*\*

All COPE Logical Systems would be eligible for *Xchange* COPE Logical System requests.

- CWXG.COPELSYS.L1

COPE Logical System L1 would be eligible for *Xchange* COPE Logical System requests.

- CWXG.COPELSYS.L2

COPE Logical System L2 would be eligible for *Xchange* COPE Logical System requests.

- CWXG.COPELSYS.L3

COPE Logical System L3 would be eligible for *Xchange* COPE Logical System requests.

Once a COPE Logical System is activated with the H line command, any user can submit jobs in that COPE Logical System for date and time exchange. There are no other security checks made by *Xchange* for jobs being exchanged under COPE Logical System support.

# Milestone 9: Assemble or Compile and Link Edit Test Programs



If your site is not using—and does not plan to use—*Xchange* test programs, you may skip this milestone and continue with [Milestone 10: Enable CICS Support](#).

The CPWR.MLXGmm.SLXGSAMP files comprise programs in various programming languages that you can use to validate *Xchange* functionality. The PDS member names suggest the languages used, and the comment box in each file briefly describes the file.



Roles involved:  
Xpediter/*Xchange* Installer

1. Compile and link the members listed in [Table 8](#) to create a system that can be used to demonstrate *Xchange* batch functionality.
2. Any of the demonstration programs listed—except XGBCHDB2—can now be used to verify that *Xchange* is functioning correctly.

**Table 8.** Test Programs

Member Name	Description
CWXGCOB	The driver program.
CWXGDATE and CWXGSUBC	Called subprograms.
CWXGDATA	Data used in the system.
CWXGJCLC	JCL to run the system.
XGBASM	An Assembler program.
XGBASMLE	A High-Level Assembler program for LE.
XGBCOBLE	A COBOL program for LE.
XGBPLI	A PLI program.
XGBPLILE	A PLI program for LE.
XGBCHDB2	A program with DB2 calls.



If your site has Xpediter/TSO or Xpediter/CICS and you want to use them with the test programs listed in this step, compile the programs with the appropriate Compuware Shared Services language processor(s). For more details, see the *Compuware Shared Services Installation and Configuration Guide*.

## CICS Transaction Support Test Programs



If your site does not require CICS transaction support, you may skip the rest of this step.

Compile and link the programs listed in [Table 9](#) for use in demonstrating *Xchange* CICS transaction support.

The resource definitions for the following programs or transactions will be added in Task 10.1 [Enable CICS Transaction Support](#).

**Table 9.** CICS Transaction Support Test Programs

Program Name or Transaction Name	Description
XGDEMCBL	A COBOL program with a transaction ID of XGCB.
XGDEMASM	The main Assembler program that links to XGLINKTD or transfers control to XGXCTLTD. The transaction ID for XGDEMASM is XGAS.
XGLINKTD	An Assembler program that links to XGLINK02.
XGLINK02	An Assembler program.
XGXCTLTD	An Assembler program.
XGCCOBLE	A COBOL program for LE. The transaction ID for XGCCOBLE is XGCL.
XGCPLI	A PLI program with a transaction ID of XGPL.
XGCPLILE	A PLI program for LE. The transaction ID is XGPE.
XGCDB2	A program with DB2 calls. XGCD is the transaction ID.

## IMS Message Level Support Test Programs



If your site does not require IMS Message Level Support, you may skip the rest of this step.

The test programs listed in [Table 10](#) are provided for use in demonstrating *Xchange* IMS Message Level Support.

**Table 10.** IMS Message Level Support Test Programs

Program Name and Transaction Name	Language Application
XGIASM	Assembler
XGICOB	COBOL (non-LE)
XGIPLI	PL/I
XGICOBLE	COBOL (LE)

To set up these test programs, perform the following steps:

1. Define the *Xchange* IMS test programs and their associated transactions to IMS.
  - a. Edit the definitions found in CPWR.MLXG $n$ nn.SLXGDATA member XGIDEFT, substituting the desired message class codes for each occurrence of **xx**, then add those definitions to the appropriate section of your site's IMS STAGE 1 SYSGEN input stream.



- b. Perform an IMS CTLBLKS gen.
2. Perform PSBGENS for the *Xchange* IMS test programs using your site's standard PSBGEN utility JCL. Use CPWR.MLXGmm.SLXGDATA member XGIDFP as input to the PSBGEN utility.
3. Generate the appropriate ACB control blocks for the *Xchange* IMS test programs using your site's standard ACBGEN utility.
4. Run the IMS online change utility to load the control blocks you have generated into the inactive IMS system libraries. These control blocks are ACBLIB and MODBLKS.
5. Assemble and compile the IMS test programs using your site's standard JCL.
6. Use CPWR.MLXGmm.SLXGDATA member XGIMFS as the MFS source to your site's standard JCL for running an MFSGEN. Run this job and verify that the MFS output is correctly created.
7. Recycle the IMS system and/or execute the necessary /MODIFY PREPARE and /MODIFY COMMIT commands to activate your new control blocks.



Sites with IMS must perform the tasks in [Milestone 12: Enable IMS Support](#) before using the *Xchange* demonstration programs listed in [Table 10](#).



# Milestone 10: Enable CICS Support



If your site is not using—and does not plan to use—*Xchange's* CICS support, skip this milestone and continue with [Milestone 11: Enable DB2 Support](#).

This milestone contains tasks for enabling *Xchange's* CICS transaction support and CICS region support.



Roles involved:  
Xpediter/*Xchange* Installer

- You can exchange the date and time for a CICS region using *Xchange's* CICS region support feature.
- You can exchange the date and time for individual programs, terminals, user IDs, and transactions within a CICS region using *Xchange's* CICS transaction support feature.



**Do not use *Xchange's* CICS region support and CICS transaction support at the same time. This combination can cause unpredictable results.**

**When using CICS region support, transaction and/or program exclusion lists do not apply.**

## Task 10.1 Enable CICS Transaction Support



If your site does not require CICS transaction support, skip this step and continue with [Enable CICS Region Support](#) on page 54.

You can exchange the date and time for individual programs, terminals, user IDs, and transactions within a CICS region using *Xchange's* CICS transaction support feature.

1. Add the load library CPWR.MLXGnm.SLXGLOAD containing Xpediter/*Xchange* to the DFHRPL concatenation in your site's CICS startup JCL.
2. Use Resource Definition Online (RDO) to update your site's CICS tables with CEDA transactions, either one at a time or using the RDO batch utility. For CICS TS 3.1 and above, use member XGCRDOTE in the CPWR.MLXGnm.SLXGDATA library as your RDO batch utility input dataset. Make any necessary changes, then either submit your RDO batch utility JCL to batch the required CEDA transactions, or run the transactions individually.

### Task 10.1.1 Enable Automatic Request Deletion

Normally, an Xpediter/*Xchange* user will delete their requests before leaving CICS. But if a terminal is disconnected, signed off, or logged off while an *Xchange* session is still active, the ID of that terminal will remain assigned to the abandoned (but still active) requests. Because another terminal logging on could be assigned the same ID, it is possible for a user to be assigned active *Xchange* requests they

don't know about. The first user may have left the session with requests set and no sure way of re-accessing the same session to delete them.

To prevent this from happening, *Xchange* can be set up to automatically delete requests when the terminal is signed off, logged off, and/or disconnected. This is done by modifying the exit DFHZATDX or whatever terminal autoinstall exit is being used. Then when a terminal is logged off, the modified exit will delete any *Xchange* requests owned by that terminal.

Enable automatic request deletion as follows:

1. Locate the exit DFHZATDX (or the VTAM terminal autoinstall exit used at your site) for modification. If your site uses a different autoinstall exit, its name can be found in the SIT parameter AIEXIT.
2. Add the following line of code to DFHZATDX in the "Delete Processing Section" after the IBM "Put Delete Code Here" comment:

```
EXEC CICS START INTERVAL (0) TRANSID ('XGTM') FROM (DELETE_TERM_ID)
```



Consider adding "NOHANDLE"—or other appropriate condition handling—for exception conditions such as TRANSIDERR.

3. If your site uses MRO, additional lines of code are required for each application owning region (AOR) where *Xchange* is installed. The lines should be added directly after the line shown above. Select the lines appropriate for your site from the following:

```
EXEC CICS START INTERVAL (0) *
      TRANSID('XGTn') FROM(DELETE_TERM_ID) NOHANDLE
```

where **XGTn** is the terminal owning region (TOR) transaction ID for XGTM in the AOR, or

```
EXEC CICS START INTERVAL (0) *
      TRANSID('XGTM') SYSID('aor-id') FROM(DELETE_TERM_ID) NOHANDLE
```

where **aor-id** is the SYSID of the AOR.



A continuation character must be included in position 72 of the first line, as shown.

4. Reassemble and link edit DFHZATDX.

## Task 10.1.2 Specify Program and Transaction Exclusions

*Xchange* is designed to exclude certain default programs and transactions from date and time exchanges. You can specify additional programs or transactions by editing program XGTEXCL found in the CPWR.MLXGmm.SLXGDATA PDS. Excluding a program or transaction does not prevent a user from specifying it in a request. When a match for the specified program or transaction is encountered, *Xchange* returns the actual date and time rather than substituting a value.



Program XGTEXCL is for CICS transaction support only.

1. Add an XGTXCLP entry for each program to be excluded and an XGTXCLT entry for each transaction to be excluded. The XGTXCLP entry is required whenever either entry is added and must always precede the XGTXCLT entry. An asterisk wildcard character can be used to create generic program or transaction entries as shown in this example:

```
XGTXCPL 'PROG*'
XGTXCCL 'TRN*'
```

2. When no programs but one or more transactions are to be excluded, be sure to include the XGTXCPL entry as shown here:

```
XGTXCPL
XGTXCCL 'TRN*'
```

3. When no transactions but one or more programs are to be excluded, be sure to include the XGTXCCL entry as shown here:

```
XGTXCPL 'PROG*'
XGTXCCL
```

4. After adding the desired entries, perform the following:
  - a. Assemble XGTEXCL
  - b. Relink XGTMAIN and XGTSEQL
  - c. Newcopy XGTMAIN and XGTSEQL
  - d. STOP *Xchange* CICS interface - STOP primary command on XGTS line
  - e. Enter XGTS/XGTM to restart CICS *Xchange* interface

Sample JCL for assembling XGTEXCL and relinking XGTMAIN and XGTSEQL is provided in CPWR.MLXGmm.SLXGCNTL PDS member ASMXCL.



If changes are later made to XGTXCCL, [Step 4](#) above must be performed again.

### Task 10.1.3 Enable PLT Initialization

You may use PLT initialization to enable *Xchange* CICS transaction support and set *Xchange* CICS transaction support requests as follows:

1. To enable *Xchange* PLT initialization, add:

```
DFHPLT TYPE=ENTRY,PROGRAM=XGTSEQL
```

to your PLTPI table, placing it after the DFHDELIM entry.

2. Define a PDS or PDSE for DD XGSEQINP to hold *Xchange* CICS transaction support requests with the following attributes:
  - Record format: FB
  - Record length: 80.

Sample JCL to define a PDS or PDSE is provided in CPWR.MLXGmm.SLXGCNTL PDS member DEFSEQL.

3. Create records in the input dataset as described in the *Xpediter/Xchange User Guide* in the section entitled "Using CICS Transaction Support with Input from a PDS or PDSE".
4. Add the input DD name to your site's CICS startup JCL as described in the *Xpediter/Xchange User Guide* in the section entitled "Using CICS Transaction Support with Input from a PDS or PDSE".

### Task 10.1.4 Enable PLT Shutdown

You may use the CICS PLT shutdown to free up the storage used for *Xchange* requests and messages. If program XGTMAIN is included in your PLT shutdown table, all *Xchange* requests and messages will be deleted, and their storage freed, when the CICS region is brought down.

To enable *Xchange* PLT shutdown, add

```
DFHPLT TYPE=ENTRY,PROGRAM=XGTMAIN
```

to your PLTSD table, placing it before the DFHDELIM entry.

## Task 10.2 Enable CICS Region Support



If your site does not require CICS region support, skip this step and continue with [Milestone 11: Enable DB2 Support](#).



**Do not use *Xchange's* CICS region support and CICS transaction support at the same time. This combination can cause unpredictable results.**

**When using CICS region support, transaction and/or program exclusion lists do not apply.**

### Using STCKSYNC Support

STCKSYNC support is required with CICS region support. The activation of STCKSYNC support is done during the startup of *Xchange*. By enabling the *Xchange* SVC, you also enable STCKSYNC support. See [Milestone 7: Enable \*Xchange\* SVCs](#) for additional information.

### Task 10.2.1 Enable PLT Initialization

Due to the way in which the CICS ABSTIME value is returned, *Xchange* must be started in the PLT.

Enable *Xchange* PLT initialization as follows:

1. Add the following to your PLTPI table, placing it after the DFHDELIM entry:

```
DFHPLT TYPE=ENTRY ,PROGRAM=XGTREGN
```

2. Use Resource Definition Online (RDO) to update your site's CICS tables with CEDA transactions in member XGCRDOR of dataset CPWR.MLXG $nnn$ .SLXGDATA, either one at a time or using the RDO batch utility.
3. Copy load modules XGTREGN and XGTABSTM from dataset CPWR.MLXG $nnn$ .SLXGLOAD to a load library in the DFHRPL of your site's CICS startup JCL.

### Task 10.2.2 Enable Midnight Rollover

*Xchange's* CICS region support requires additional midnight rollover processing. To enable this support, perform the following steps:

1. Add the following line to your PLTPI table after the DFHDELIM entry:

```
DFHPLT TYPE=ENTRY ,PROGRAM=XGTAJP3
```



If you completed [Enable CICS Transaction Support](#) on page 51, you may skip steps 2 and 3 below.

2. Add the load library CPWR.MLXG $nnn$ .SLXGLOAD containing *Xchange* to the DFHRPL concatenation in your site's CICS startup JCL.
3. Use Resource Definition Online (RDO) to update your site's CICS tables with CEDA transactions, either one at a time or using the RDO batch utility. For CICS TS 3.1 and above, use member XGCRDOTE in the CPWR.MLXG $nnn$ .SLXGDATA library as your RDO batch utility input dataset. Make any necessary changes, then either submit your RDO batch utility JCL to batch the required CEDA transactions, or run the transactions individually.

# Milestone 11: Enable DB2 Support



If your site is not using—and does not plan to use—*Xchange's* DB2 support, you may skip this milestone and continue with [Milestone 12: Enable IMS Support](#).

This milestone contains tasks for enabling DB2 support.



Roles involved:  
Xpediter/*Xchange* Installer

## Overview

*Xchange* DB2 support provides date/time simulation for SQL time functions while preserving the integrity and availability of DB2. With DB2 support enabled, *Xchange* will intercept DB2 special registers CURRENT DATE, CURRENT TIME, and CURRENT TIMESTAMP and return whatever values were specified on the Simulated Date Time Settings screen. Date, time, and timestamp columns defined as NOT NULL WITH DEFAULT will also be exchanged.

*Xchange* DB2 support is enabled by applying a USERMOD zap to the CSECT named DSNXVCTS in the IBM module DSNXGRDS. This CSECT is used solely for SQL date functions, so other DB2 dates are unaffected. The zapped module must be accessible to the DB2 database's (DBM) address space—for example in the STEPLIB. Cycling the address space may be necessary.

*Xchange* DB2 Distributed Data Facility (DDF) support requires the same USERMOD zap. The zapped module must be accessible to the DB2 DIST address space—for example in the STEPLIB. Cycling the address space may be necessary.

*Xchange* DB2 Stored Procedure support requires the same USERMOD zap. The zapped module must be accessible to the Workload Manager (WLM) address space—for example in the STEPLIB. Cycling the address space may be necessary.

If *Xchange* fails during processing, its full recovery support prevents DB2 and the user application from abending and allows DB2 to perform its normal recovery and cleanup. DB2 returns a date of all asterisks to the caller and an SQL-187 return code just as it would if *Xchange* had not been involved.

### Task 11.1 Apply the DB2 Zap

Compuware recommends that SMP/E be used to apply the *Xchange* DB2 zap. There are two advantages to using SMP/E:

- If DB2 maintenance is later applied, you will be alerted to any regressions.
- The zap can be easily removed with the SMP/E RESTORE function.

1. Select the sample jobstream in CPWR.MLXGnm.SLXGCNTL that corresponds to your site's DB2 release from the list in [Table 11](#).

**Table 11.** Xchange DB2 Zap Members

DB2 Release	Sample Jobstream	
	Non-SMP	SMP/E
6.1	XGDB2A61	XGDB2S61
7.1	XGDB2A71	XGDB2S71
8.1	XGDB2A81	XGDB2S81
9.1	XGDB2A91	XGDB2S91
10.1	XGDB2AA1	XGDB2SA1
11.1	XGDB2AB1	XGDB2SB1
12.1	XGDB2AC1	XGDB2SC1

Both jobstreams for each DB2 release point to a common zap member, XGZPDBxx, found in the CPWR.MLXGnm.SLXGDATA PDS. The xx corresponds to the final two digits of the jobstream.

2. Before applying the zap by running the appropriate job, either comment out or uncomment the EXPAND statement on the first line of XGZPDBxx as follows:
  - For SMP/E, replace the asterisk (\*) in position 1 with a blank.
  - For non-SMP, make sure an asterisk is in position 1.
3. Apply the zap.



Sites with DB2 can now use the Xchange batch demonstration program XGBCHDB2 previously compiled and link edited in [Milestone 9: Assemble or Compile and Link Edit Test Programs](#).

Sites with DB2 and CICS can now use the Xchange demonstration program XGCDB2.



# Milestone 12: Enable IMS Support



If your site is not using—and does not plan to use—*Xchange's* IMS support, you may skip this milestone and continue with [Milestone 13: Enable Xpediter/TSO Interface](#).

This milestone contains tasks for enabling IMS support.



Roles involved:  
Xpediter/*Xchange* Installer

With *Xchange's* IMS Message Level Support, you can exchange the date and time for IMS Message Processing Programs (MPPs), including Wait For Input (WFI) transactions, Interactive Fast Path (IFP) programs, and Batch Message Processing (BMP) regions.



**Do not use the ISPF Interface to set a region level pattern request for the same message processing region being used for IMS Message Level Support. This combination can cause unpredictable results.**

## Task 12.1 Enable IMS Message Level Support

To enable IMS Message Level Support, perform the following steps:

1. Define the *Xchange* Message Level Support programs and their associated transactions to IMS.
  - a. Edit the definitions found in CPWR.MLXGmm.SLXGDATA member XGIMSST1, substituting the desired message class codes for each occurrence of xx, then add those definitions to the appropriate section of your site's IMS STAGE 1 SYSGEN input stream.
  - b. Perform an IMS CTLBLKS gen.
2. Ensure that the message class codes associated with the *Xchange* transactions XGTM, XGTS, XGTMI02, XGTSI02, XGTMI03, and XGTSI03 are available in the Message Processing Region(s) (MPRs) in which they will be running.



The XGTM, XGTS, XGTMI02, XGTSI02, XGTMI03, and XGTSI03 transactions do not need to be defined in every MPR in which transactions will have their date and time exchanged. They only need to be defined in the MPR(s) where *Xchange* requests will be entered. Transactions to be exchanged can be run in any connected region.

3. Perform PSBGENs for the *Xchange* programs XGTMI01, XGTMI02, XGTMI03, XGTSI01, XGTSI02, and XGTSI03 using your site's standard PSBGEN utility JCL.
 

Use CPWR.MLXGmm.SLXGDATA members XGTMI01, XGTMI02, XGTMI03, XGTSI01, XGTSI02, and XGTSI03 as input to the PSBGEN utility.
4. Generate the appropriate ACB control blocks for programs XGTMI01, XGTMI02, XGTMI03, XGTSI01, XGTSI02, and XGTSI03 using your site's standard ACBGEN utility.
5. Generate Message Format Services (MFS) control blocks for XGTM, XGTMC, XGTMV, XGTS, XGTSC, XGTSV, XGEMA, and XGHPA running your site's MFS utility with

CPWR.MLXG $nnn$ .SLXGDATA members XGTMTM, XGTMTC, XGTMTV, XGTMTS, XGTSTC, XGTSTV, XGTMEMA, and XGTMPHA as input.



In the furnished MFS source, the DEV macro specifies TYPE=(3270,2). Some sites may require a different TYPE, such as TYPE=3270-A02. If necessary, change the TYPE operand of the DEV macro to a value appropriate for your site.

6. Run the IMS online change utility to load the control blocks you have generated into the inactive IMS system libraries.

These control blocks are ACBLIB, MODBLKS, and FMTLIB members. If you want to use IMS security to restrict access to the *Xchange* administrative user transaction XGTS, refer to the appropriate IBM IMS/ESA documentation.

7. Use IDCAMS to create each IMS request dataset.

All dependent regions that connect to a control region must share the same IMS request dataset. That dataset cannot be shared by dependent regions connected to any other control region. The JCL for creating and initializing an IMS request dataset is included in the CPWR.MLXG $nnn$ .SLXGCNTL member ALLOCIMS.

8. Edit ALLOCIMS following the instructions given near the top of the file, then submit the job.
9. Modify the JCL of each dependent region in which *Xchange* date and time simulations are to be performed as follows:
  - a. In the step that executes DFSRRC00, concatenate the *Xchange* loadlib to the region's steplib.



If Xpediter/*Xchange* will be working with LE date/time calls in an MPR that specifies a pre-load of CEELRRIN, you must have the *Xchange* zap to CEEPLPKA in the STEPLIB concatenation **before** the SCEERUN library. If you created a zapped CEEPLPKA module in [Apply the DB2 Zap](#) on page 55, you can use that same zapped CEEPLPKA module in this step. An alternative solution is to remove CEELRRIN from the pre-load list.

- b. Add the following DD statement, replacing **customer.ims.file** with the dataset name you defined in [Step 7](#) above:

```
//XGIMSREQ DD DSN=customer.ims.file,DISP=SHR
```

- c. Add the following step at the end of the region's JCL, replacing CPWR with the high-level qualifier chosen for your site and again replacing **customer.ims.file** with the dataset name you defined in [Step 7](#) above:

```
//CLEANUP EXEC PGM=XGCLNREQ,COND=EVEN
//STEPLIB DD DSN=CPWR.MLXG $nnn$ .SLXGLOAD,DISP=SHR
//XGIMSREQ DD DSN=customer.ims.file,DISP=SHR
```

10. Edit CPWR.MLXG $nnn$ .SLXGDATA member XGPCCUPD for IMS 10.1 and below, XGPCCUP1 for IMS 11.1 through IMS 13.1, or XGPCCU14 for IMS 14.1, or XGPCCU15 for IMS 15.1 and above, changing the first character on the first line from a dollar sign (\$) to a period (.).
11. If you want to use SMP/E to install the *Xchange* interface with the IMS module, go to sub[Step 12](#). Otherwise, go to sub[Step 13](#).
12. Review the JCL to perform the SMP/E install of the *Xchange* interface.

JCL to perform an SMP/E install of the *Xchange* interface with the IMS module DFSPCC20 is provided in CPWR.MLXG $nnn$ .SLXGCNTL member XGIMSS $xx$ , where  $xx$  represents the IMS release used at your site. (If using IMS release 10.1 or above, the member name is XGIMSS $xxx$ .) If necessary, this USERMOD may be removed from DFSPCC20 with the SMP/E RESTORE function.

Compuware recommends that the RESLIB DD/DDDEF for this step point to a library other than your normal IMS RESLIB. The selected library should then be included ahead of your normal IMS

RESLIB in the IMS STEPLIB concatenation of all dependent regions in which date and time exchanges are to be performed.

- a. Before submitting the job, edit the appropriate member to make any necessary changes including jobcard, SMPE PROC, and ZONE names.
- b. Submit the job.

The job should complete with an SMP/E RC4. The assembly will complete with RC4, and the LINK will complete with RC8. SMP/E messages GIM31902I, GIM38201W, and GIM67301W can be ignored.

- c. Edit member XGLNKPCC or XGLNKPC1 for IMS 8.1 and above, making any necessary JCL changes.

The SYSLIB DD statement should point to the same RESLIB used in the XGIMSSxx/XGIMSxxx job.

- d. Submit XGLNKPCC or XGLNKPC1 for IMS 8.1 and above to link module XGIMSTFR into DFSPCC20.

This job should complete with RC0.

- e. *Go to sub*[Step 15](#).

13. Find CPWR.MLXGmm.SLXGCNTL member:

- XGPCC20U for IMS releases prior to 8.1 *or*
- XGPCC20X for IMS 8.1 through IMS 10.1 *or*
- XGPCC20B for IMS 11.1 through IMS 13.1 *or*
- XGPCC20E for IMS 14.1 *or*
- XGPCC20F for IMS 15.1 and above.

It contains JCL to install the *Xchange* interface with IMS module DFSPCC20.

14. Before submitting the job, edit the XGPCC20U, XGPCC20X, XGPCC20B, XGPCC20E, or XGPCC20F member selected in the previous step, changing the DSNs to conform to your site's standards and making the changes required in the following three job steps:

- a. Use IEBUPDTE to create a temporary copy, SYSUT2, of IMS module DFSPCC20.

SYSUT1 must point to the dataset containing the current source code for DFSPCC20.

- b. Assemble the updated temporary copy of DFSPCC20 created in [Step a](#).
- c. Link edit the new DFSPCC20 object module created in [Step b](#).

Compuware recommends that the SYSLMOD DD statement for this step point to a library other than your normal IMS RESLIB. The selected library should then be included ahead of your normal IMS RESLIB in the IMS STEPLIB concatenation of all dependent regions in which date and time exchanges are to be performed.

15. Ensure your site's MPP jobnames are defined as jobs to *Xchange* external security as described in [Milestone 8: Configure Xchange External Security](#).

The user who submits the job that starts the MPP must have update access to the *Xchange* pseudo-dataset name.

16. Recycle the IMS system and execute the necessary /MODIFY PREPARE and /MODIFY COMMIT commands to activate your new control blocks.



Sites with IMS can now use the *Xchange* demonstration programs listed in [Table 10](#) in [Milestone 9: Assemble or Compile and Link Edit Test Programs](#).



## Milestone 13: Enable Xpediter/TSO Interface



If your site is not using—and does not plan to use—*Xchange's* Xpediter/TSO interface, you may skip this milestone and continue with [Milestone 14: Enable Natural Support](#).

This milestone configures Xpediter/*Xchange* to allow users to invoke it from within an Xpediter/TSO session by typing XCHANGE on the COMMAND line and pressing Enter. Syntax and options for using the XCHANGE command are discussed in the *Xpediter/Xchange User Guide*.



Roles involved:  
Xpediter/*Xchange* Installer

### Task 13.1 Make Libraries Accessible to Xpediter/*Xchange* Users

The *Xchange* SYSPROC and ISPF libraries can also be made accessible to *Xchange* users via their TSO logon CLIST. If you created new libraries for *Xchange*, allocate those new libraries before ISPF initiation by adding the following allocations to the logon CLISTs:

```
ALLOC F(SYSPROC) DA('h1q.SLXGCLIB') SHR
ALLOC F(ISPMLIB) DA('h1q.SLXGMSG') SHR
ALLOC F(ISPPLIB) DA('h1q.SLXGPNL') SHR
ALLOC F(ISPLLIB) DA('h1q.SLXGLOAD') SHR
```

### Task 13.2 Update the Xpediter/TSO LIBDEF

You must also add allocation LIBDEF statements for the datasets listed in [Table 12](#) to your Xpediter/TSO startup CLIST:

**Table 12.** Xpediter/TSO CLIST Allocations

Allocation	Dataset
ISPPLIB	CPWR.MLXGnnn.SLXGPENU
ISPMLIB	CPWR.MLXGnnn.SLXGMENU
ISPLLIB	CPWR.MLXGnnn.SLXGLOAD



# Milestone 14: Enable Natural Support



If your site is not using—and does not plan to use—*Xchange's* Natural support, you may skip this milestone and continue with [Milestone 15: Enable COPE Logical System Support](#).

This milestone configures *Xpediter/Xchange* to support the Natural application development and deployment environment from Software AG.



Roles involved:  
*Xpediter/Xchange* Installer

## Task 14.1 Enable Natural Support

To provide support for Natural date/time requests, *Xpediter/Xchange* supplies its own version of the Natural CMCOTIME exit. The *Xchange* version of the CMCOTIME exit must be linked into the Natural nucleus and enabled as follows:

1. The *Xpediter/Xchange* version of the CMCOTIME exit is named XGCOTIME and is located in the CPWR.MLXGnnn.SLXGLOAD library. As shown in [Figure 4](#), an INCLUDE card pointing to the *Xchange* load library and the XGCOTIME member is required for the re-linking of the Natural nucleus.
2. Set the Natural TD parameter to TD=0.

**Figure 4.** *Xchange* Link Edit Cards to Re-Link the Natural Nucleus

```
// JCL required
//   to re-link the
//     Natural nucleus
//XGLIB DD DISP=SHR,DSN=CPWR.MLXGnnn.SLXGLOAD
//SYSIN DD *
INCLUDE XGLIB(XGCOTIME)
(REST OF THE LINK EDIT CARDS
 TO RE-LINK THE NATURAL NUCLEUS)
```





# Milestone 15: Enable COPE Logical System Support



If your site is not using—and does not plan to use—*Xchange's* COPE Logical System support, you may skip this milestone and continue with [Milestone 16: Deployment](#).

This milestone configures *Xpediter/Xchange* to support the IMS Logical Systems created by Compuware's COPE product.



Roles involved:  
*Xpediter/Xchange* Installer

## Task 15.1 Enable COPE Logical System Support

With *Xchange's* COPE Logical System support, you can exchange the date and time for IMS Message Processing Programs (MPPs), including Wait for Input (WFI) transactions, Interactive Fast Path (IFP) programs, and Batch Message Processing (BMP) regions at the COPE Logical System level.



**If an individual *Xchange* request is entered on the primary Simulated Date Time Settings screen for a job that happens to run in a COPE Logical System for which there is an active COPE Logical System request, the date/time fields from the individual request will be used instead of the COPE Logical System screen's date/time fields. This is because specific individual requests always take precedence over COPE Logical System requests.**

1. To enable COPE Logical System support for IMS DC, copy the following modules from COPE.LOAD to your COPE RESLIB that resides in the STEPLIB concatenation of all COPE message processing regions in which date and time exchanges are to be performed:
  - COPEGLSY
  - COPETRA
2. Recycle the affected message processing regions.
3. To enable COPE Logical System support for IMS DB, copy the following modules from COPE.LOAD to your COPE RESLIB that resides in the STEPLIB concatenation of your batch jobs in which date and time exchanges are to be performed:
  - COPEGLSY
  - COPETRA
4. Add a COPEBSYS DD to your batch jobs to identify the Logical System to *Xchange* as follows:

```
//COPEBSYS DD UNIT=SYSDA,SPACE=(TRK,1),DSN=&&7sys
```

where  $l_{sys}$  is a value specified on an *Xchange* COPELSYS parameter.



Do not identify the Logical System in the second positional JOB card parameter or in place of the IMSID in the PARM field of the DFSRRC00 program.

# Milestone 16: Deployment

This chapter contains considerations to be aware of and tasks to perform during deployment of *Xpediter/Xchange*.



*Xpediter/Xchange* Installer

## Task 16.1 Target Library Deployment

Copy the applicable libraries listed in [Table 2](#) to the destination LPAR.



# Checklist of Milestones and Tasks

- ❑ Milestone 1: Ensure Installation and Configuration of Companion Products
  - ❑ Task 1.1 Install/Upgrade Enterprise Common Components
  - ❑ Task 1.2 Apply ECC Maintenance
  - ❑ Task 1.3 Import Xpediter/*Xchange* License
- ❑ Milestone 2: Install Xpediter/*Xchange* Using SMP/E
  - ❑ Task 2.1 Ensure Product Integrity
  - ❑ Task 2.2 Follow the Compuware Installation Guide
- ❑ Milestone 3: Specify Xpediter/*Xchange* Properties and Parameters
  - ❑ Task 3.1 Update the MVS Program Properties Table
  - ❑ Task 3.2 Implement the Compuware PARMLIB
    - ❑ Task 3.2.1 Create the Default Member
    - ❑ Task 3.2.2 Specify Xpediter/*Xchange* Parameters
    - ❑ Task 3.2.3 Update the CMSC with your PARMLIB Members
- ❑ Milestone 4: Update TSO CLISTS
  - ❑ Task 4.1 Copy Members
  - ❑ Task 4.2 Edit the CLISTS
- ❑ Milestone 5: Allocate Journal and Save/Restore Datasets
  - ❑ Task 5.1 Allocate the Journal Datasets
  - ❑ Task 5.2 Allocate the Save/Restore Dataset
- ❑ Milestone 6: Enable Language Environment (LE) Support
  - ❑ Task 6.1 Enable LE for CICS Region Support
    - ❑ Task 6.1.1 Apply CEEPLPKA Zap

- ❑ Task 6.2 Enable LE for DB2 Support
- ❑ Task 6.3 Enable Enterprise PL/I DATE or DATETIME Built-in Function Support
- ❑ Task 6.4 Enable C time() and gettimeofday64() Function Support
- ❑ Milestone 7: Enable *Xchange* SVCs
  - ❑ Task 7.1 Enable SVC for Enhanced STCK and STCKSYNC Macro Support
  - ❑ Task 7.2 Enable SVC for Enhanced STCKE and STCKE Macro Support
- ❑ Milestone 8: Configure *Xchange* External Security
  - ❑ Task 8.1 Assemble and Link the RACROUTE Module
  - ❑ Task 8.2 CA-TOP SECRET and ACF2 Configuration
  - ❑ Task 8.3 Define Security Rules
    - ❑ Task 8.3.1 Establish User ID Access
    - ❑ Task 8.3.2 Define Security Rules for Jobclass Support
    - ❑ Task 8.3.3 Define Security Rules for DB2 Distributed Data Facility (DDF) Support
    - ❑ Task 8.3.4 Establish Resource Rules
    - ❑ Task 8.3.5 Designate *Xchange* Administrators
    - ❑ Task 8.3.6 Define Security Rules for COPE Logical System Support
- ❑ Milestone 9: Assemble or Compile and Link Edit Test Programs
- ❑ Milestone 10: Enable CICS Support
  - ❑ Task 10.1 Enable CICS Transaction Support
    - ❑ Task 10.1.1 Enable Automatic Request Deletion
    - ❑ Task 10.1.2 Specify Program and Transaction Exclusions
    - ❑ Task 10.1.3 Enable PLT Initialization
    - ❑ Task 10.1.4 Enable PLT Shutdown
  - ❑ Task 10.2 Enable CICS Region Support
    - ❑ Task 10.2.1 Enable PLT Initialization
    - ❑ Task 10.2.2 Enable Midnight Rollover
- ❑ Milestone 11: Enable DB2 Support
  - ❑ Task 11.1 Apply the DB2 Zap

- ❑ Milestone 12: Enable IMS Support
  - ❑ Task 12.1 Enable IMS Message Level Support
- ❑ Milestone 13: Enable Xpediter/TSO Interface
  - ❑ Task 13.1 Make Libraries Accessible to Xpediter/Xchange Users
  - ❑ Task 13.2 Update the Xpediter/TSO LIBDEF
- ❑ Milestone 14: Enable Natural Support
  - ❑ Task 14.1 Enable Natural Support
- ❑ Milestone 15: Enable COPE Logical System Support
  - ❑ Task 15.1 Enable COPE Logical System Support
- ❑ Milestone 16: Deployment
  - ❑ Task 16.1 Target Library Deployment

