



# **SHADOW MAINFRAME ADAPTER CLIENT FOR NATURAL**

**SHADOW MAINFRAME ADAPTER SERVER INSTALLATION  
SHADOW INTERFACE FOR NATURAL INSTALLATION**

***POWERED BY***  
***SHADOW***

January, 2004

This document is published by the NEON Systems, Inc. Technical Publications Department and applies to Shadow Mainframe Adapter Client for Natural.

Copyright © 1994-2003 NEON Systems, Inc. All rights reserved. Printed in the U.S.A.

Licensee is granted permission to make a limited number of copies of the documentation for its internal business purposes only. All such copies shall bear all copyright, trade secret, trademark and any other intellectual property notices on the original copies. This limited right to reproduce for internal purposes only is not transferable. Furthermore, this limited right DOES NOT include any license to distribute, modify, display or make derivative works from the Copyrighted materials.

NEON, Shadow, Shadow Direct, and Enterprise Direct are registered trademarks, and the NEON logo, Shadow Activity Monitor, Shadow Advanced Controls, Shadow Advanced Scalability, Shadow AutoHTML, Shadow Mainframe Adapter Client, Shadow Enterprise Auditing, Shadow Enterprise Direct, Shadow Enterprise Transactions, Shadow Event Facility, Shadow Enterprise Transactions, Shadow Interface, Shadow JDBC Adapter, Shadow MDI Replacement Module, Shadow REXX/Tools, Shadow Mainframe Adapter Server, Shadow SSL Support Module, Shadow Support Module, Shadow Web Interface, and Shadow Web Server are trademarks of NEON Systems, Inc. in the USA and in other select countries.

The symbols ® and ™ denote USA trademark rights.

All other trademarks are the property of their respective owners.

Throughout this publication, NEON Systems, Inc. is also, for convenience, referred to as "NEON." The Reader should not presume that such use of NEON conflicts with the use of NEON as a registered trademark associated with certain products of NEON Systems, Inc.

This software/documentation contains proprietary information of NEON Systems, Inc.; it is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited.

If this software/documentation is delivered to any U.S. Government Agency, then it is delivered with Restricted Rights and the following legend is applicable:

#### **Restricted Rights Legend**

Use, duplication, or disclosure by the U.S. Government is subject to restrictions set forth in FAR Section 52.227-14 (June 1987) Alt. III(g)(3)(June 1987), FAR Section 52.227-19 (June 1987), or sub-clause (c)(1)(ii) of Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, as applicable.  
Contractor is NEON Systems, Inc. 14100 Southwest Freeway, Suite 500, Sugar Land, Texas 77478.

NEON Systems, Inc. does not warrant that this document is error-free. The information in this document is subject to change without notice and does not represent a commitment on the part of NEON Systems, Inc. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without the express written permission of an authorized representative of NEON Systems, Inc.

Address inquiries to:

**NEON Systems, Inc.**

14100 SW Freeway, Suite 500  
Sugar Land, Texas 77478

World Wide Web: <http://www.neonsys.com>

Phone: 1-800-505-6366  
(281) 491-4200 (Corporate Sales, Customer Support)  
Fax: (281) 242-3880



# Contents

---

## Part I: Introduction

|   |            |
|---|------------|
| <b>Chapter 1: Introduction: Shadow Mainframe Adapter Client for Natural</b> . . . . . | <b>1-1</b> |
| Overview . . . . .  | 1-1        |
| Shadow Mainframe Adapter Client for Natural . . . . .                                 | 1-1        |

## Part II: Shadow Mainframe Adapter Server Installation

|  |            |
|--|------------|
| <b>Chapter 2: Shadow Mainframe Adapter Server: Planning and Installation</b> . . . . . | <b>2-1</b> |
| Planning for the Basic Installation of Shadow Mainframe Adapter Server. . . . .        | 2-1        |
| System Requirements . . . . .  | 2-1        |
| Space Requirements . . . . .   | 2-2        |
| Distribution Package . . . . .   | 2-2        |
| Installation Features . . . . .  | 2-4        |
| Product Installation Steps. . . . .  | 2-4        |
| Step 1: Unload the CNTL Dataset. . . . .   | 2-4        |
| Step 2: Modify and Execute the Install Member. . . . .                                 | 2-5        |
| Step 3: APF-Authorize the Load Library. . . . .  | 2-5        |
| Step 4: Define new Slip Traps. . . . .   | 2-6        |
| Step 5: Create the Trace VSAM Datasets. . . . .  | 2-7        |
| Step 6: Set Up the Started Task JCL. . . . .   | 2-8        |
| Step 7: Define the Started Task Name to Your Security Product. . . . .                 | 2-10       |
| Step 8: Provide VTAM Definitions. . . . .  | 2-12       |
| Step 9: Customize the Initialization EXEC. . . . .                                     | 2-14       |
| Step 10: Set Up the ISPF/SDF Dialogs. . . . .  | 2-23       |
| Step 11: Start the Shadow Mainframe Adapter Server. . . . .                            | 2-23       |
| Step 12: Ensure that the Shadow Mainframe Adapter Client Has Been Installed. . . . .   | 2-24       |

## **Chapter 3: Shadow Mainframe Adapter Server: Authorizing Access To Resources** **3-1**

|  |                     |
|--|---------------------|
| Overview . . . . .                                   | 3-1                 |
| Protected Resources . . . . .                        | 3-2                 |
| How Resource Access Is Determined . . . . .          | 3-3                 |
| Defining Shadow Resources to RACF. . . . .           | 3-5                 |
| Defining Shadow Resources to CA-Top Secret. . . . .  | 3-6                 |
| Defining Shadow Resources to ACF2 . . . . .          | 3-7                 |
| Defining Shadow ISPF Load Modules . . . . .          | 3-7                 |
| <a href="#">Using the RACF Pass Ticket</a> . . . . . | <a href="#">3-8</a> |
| Note on Started Task Security . . . . .              | 3-9                 |

---

|  |            |
|--|------------|
| Controlling Information Access with the TRACEDATA Resource . . . . .             | 3-10       |
| Resource Security for Test Versions of Shadow Mainframe Adapter Server . . . . . | 3-10       |
| <b>Chapter 4: Shadow Mainframe Adapter Server: Connecting to TSO . . . . .</b>   | <b>4-1</b> |
| Setting Up Shadow Mainframe Adapter Server to Run Under TSO . . . . .            | 4-1        |
| Running a Test Version . . . . .   | 4-2        |

## **Part III: Shadow Interface for Natural Installation**

|   |            |
|---|------------|
| <b>Chapter 5: Shadow Interface for Natural: Planning and Installation . . . . .</b> | <b>5-1</b> |
| Installation Prerequisites . . . . .  | 5-1        |
| Installation Steps . . . . .  | 5-1        |
| Installation for CICS . . . . .   | 5-1        |
| Installation for Batch . . . . .  | 5-10       |

# About this Publication

---

This book contains planning and installation information for the Shadow Mainframe Adapter Server and the Shadow Interface™ for Natural, the server components of the Shadow product.

## How this Publication Is Organized

This book contains the following information:

### Part I: Introduction

- Chapter 1, “Introduction: Shadow JDBC Connect for Natural,” provides a brief overview of Shadow, including details about Shadow Mainframe Adapter Client for Natural.

### Part II: Shadow Server Installation

- Chapter 2, “Shadow Server: Planning and Installation,” provides planning and installation information for the Shadow Mainframe Adapter Server.
- Chapter 3, “Shadow Server: Authorizing Access To Resources,” offers instructions for authorizing access to Shadow Mainframe Adapter Server resources.
- Chapter 4, “Shadow Server: Connecting to TSO,” covers the steps required to set up the Shadow Mainframe Adapter Server to run under TSO.

### Part III: Shadow Interface for Natural Installation

- Chapter 5, “Shadow Interface for Natural: Planning and Installation,” describes the planning and installation required for the Shadow Interface for Natural.

## Reader’s Comments

Please e-mail any comments or questions you have about our documentation to [support@neonsys.com](mailto:support@neonsys.com).

Thank you!

---

# Introduction

---





# CHAPTER 1:

# ***Introduction: Shadow Mainframe Adapter Client for Natural***

---

---

This chapter gives a general introduction to the Shadow product, including details about the Shadow Mainframe Adapter Client for Natural component.

Topics include the following:

- Overview
  - Shadow Mainframe Adapter Client for Natural

## **Overview**

Shadow is an efficient, easy-to-use, and flexible solution for integrating mainframe data sources and transaction environments to client/server and n-tier environments. The unique Shadow architecture provides maximum flexibility with minimal impact on CPU cycles.

### ***Shadow Mainframe Adapter Client for Natural***

The Shadow product offers various connectivity options, including Shadow Mainframe Adapter Client for Natural. With Shadow Mainframe Adapter Client for Natural, any JDBC enabled application can use standard JDBC facilities to make requests, wrapped by a CALL statement, directly to Natural programs. The end result is a returned relational result set from the Natural application running in its native TP environment.

Shadow Mainframe Adapter Client for Natural consists of the following components:

- Shadow Mainframe Adapter Server
- Shadow Mainframe Adapter Client
- Shadow Interface™ for Natural

### **Shadow Mainframe Adapter Server**

The Shadow Mainframe Adapter Server component, which resides on the mainframe, offers the following benefits:

- Provides native access to ADABAS, CICS, DB2, IMS/DB, IMS/TM, Natural, and VSAM from a single tool.
- Eliminates of the need for a mid-tier gateway.
- Installs in less than one day.

- Incorporates centralized online monitoring, control, and diagnostic capabilities.

## Shadow Mainframe Adapter Client

The Shadow Mainframe Adapter Client component, which consists of Shadow Mainframe Adapter Client, is a connector that enables Java applications to integrate z/OS data and transactional sources through the JDBC API. Benefits include the following:

- Takes advantage of Java capabilities including multi-threading, connection pooling, and batch updates.
- Supports JVM 1.2 (J2EE) and Java servlets and is JDBC 2.0 compliant.
- Runs on a growing range of platforms including HP-UX, IBM AIX, Linux, Linux/390, Sun Solaris, and Windows.
- Performs data and SQL dialect conversations, data compression, and network optimization in conjunction with the Shadow Mainframe Adapter Server.

## Shadow Interface for Natural

The Shadow Interface for Natural allows programmers to develop client applications that can execute backend services in their Natural transaction processing (TP) environments. In this case, the Shadow Interface for Natural is providing access to Natural transactions in a CICS or batch environment. Natural is a fourth-generation language for application development.



### **Note:**

The terms “service” and “server” are used interchangeably in this documentation to refer to the remote backend application.

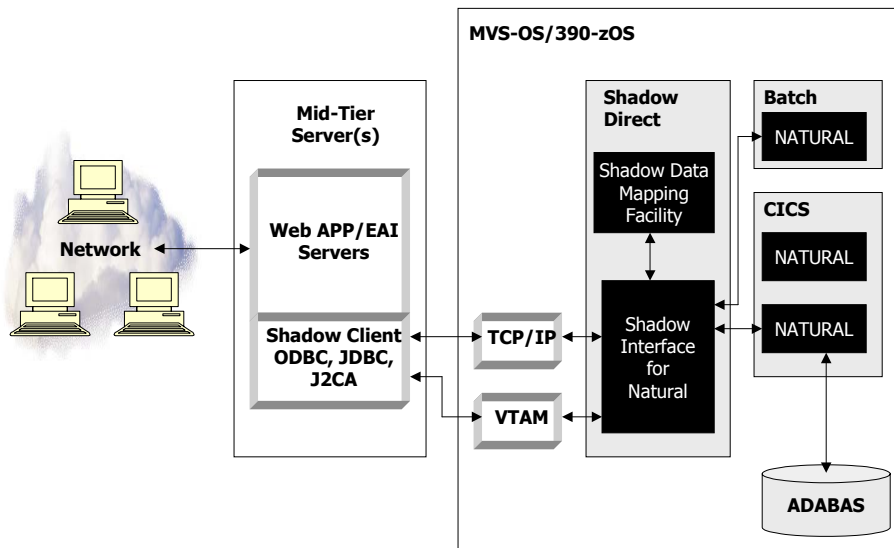
## Key Features

The Shadow Interface for Natural offers the following features:

- Automatically starts and terminates servers according to demand.
- Allows less used servers to “die,” freeing up storage for more frequently used servers and increasing available resources.
- Limits the lifespan of the Natural server, preventing application memory leaks from reaching a critical point.
- Provides EXCI connection failover, so that if the primary CICS system goes down, Shadow will automatically start the server on a backup CICS.

## Architecture

Figure 1–1 illustrates the architecture for the Shadow Interface for Natural.



**Figure 1–1. Architecture for the Shadow Interface for Natural**

The Shadow Interface for Natural was designed as an alternative to Software AG’s EntireX Broker Advanced Communication Interface (ACI) component. The application programming interface (API) between the Shadow Interface for Natural and Natural is identical to that of the EntireX implementation and function.

Unlike similar products, the Shadow Interface for Natural can be used in a 2-tier Web Server solution and offers all the benefits and advantages offered by Shadow, including trace browse, security, scalability, EXCI failover, and much more. It also provides resource protection and metadata support.

### **Natural Send/Receive Process**

The Shadow Interface for Natural can start and manage application services in a CICS region. When a client requests a service to be executed, this module searches for an available instance of the service.

If none are available, or if none have been previously started, the Shadow Interface for Natural determines, based on the Shadow Data Mapping Facility (DMF) information for the service, the TP environment in which the application resides. It then starts a copy of the application in that TP environment. Once started, the application service must register itself to the Shadow Mainframe Adapter Server.

After it is registered, the service application can accept a work request (receive), process it, and then return the results back to the client (send). The service can then wait for another unit of work from the same or different client. This process continues until the Shadow Interface for Natural or the service application determines that the service is no longer needed. The service program will then deregister itself with the Shadow Mainframe Adapter Server, and terminate.

Figure 1–2 shows how the Shadow Interface for Natural receives and sends messages.

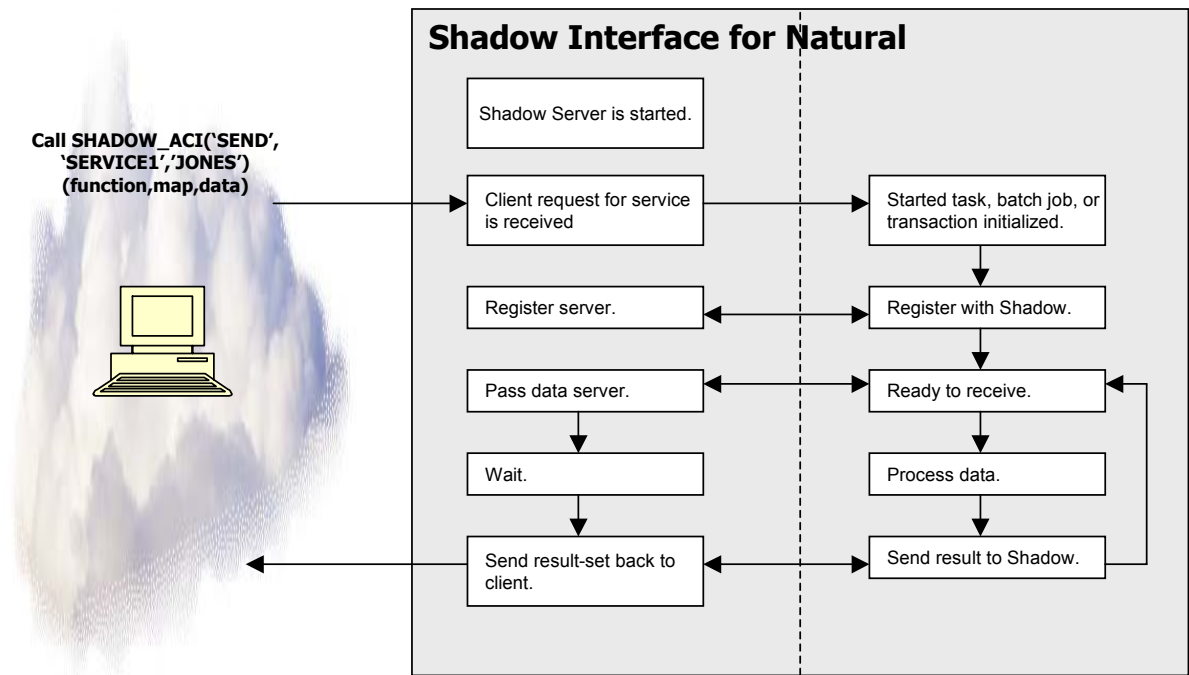


Figure 1–2. The Shadow Interface for Natural Receive/Send Process

# Shadow Mainframe Adapter Server Installation

---



## CHAPTER 2:

# **Shadow Mainframe Adapter Server: Planning and Installation**

---

---

This chapter covers planning considerations and installation steps for the Shadow Mainframe Adapter Server, the server component of the Shadow product.

Topics include:

- Planning for the Basic Installation of Shadow Mainframe Adapter Server
  - System Requirements
  - Space Requirements
  - Distribution Package
- Installation Features
- Product Installation Steps
  - Step 1: Unload the CNTL Dataset.
  - Step 2: Modify and Execute the Install Member.
  - Step 3: APF-Authorize the Load Library.
  - Step 4: Define new Slip Traps.
  - Step 5: Create the Trace VSAM Datasets.
  - Step 6: Set Up the Started Task JCL.
  - Step 7: Define the Started Task Name to Your Security Product.
  - Step 8: Provide VTAM Definitions.
  - Step 9: Customize the Initialization EXEC.
  - Step 10: Set Up the ISPF/SDF Dialogs.
  - Step 11: Start the Shadow Mainframe Adapter Server.
  - Step 12: Ensure that the Shadow Mainframe Adapter Client Has Been Installed.

## **Planning for the Basic Installation of Shadow Mainframe Adapter Server**

This section will cover the following planning considerations for the basic installation of Shadow Mainframe Adapter Server:

- System Requirements.
- Space Requirements.
- Distribution Package
- Installation Requirements

## System Requirements

Shadow Mainframe Adapter Server requires the following host software:

- MVS/ESA (any level).
- VTAM 3.2 or later.
- TSO/E version 2 or later.
- ISPF 2.3 or later.
- (If accessing DB2 or using features requiring DB2) DB2 version 2 or later.
- Any IBM-supported release of RACF, ACF2 release 4.1 or later, or Top Secret.

## Space Requirements

Shadow Mainframe Adapter Server requires space equal to about 5,567 tracks.

## Distribution Package

Before beginning installation, verify that you received a complete distribution package. The package should contain the following items:

- The following CDs:
  - Shadow Mainframe Adapter Client CD.
  - Shadow Documentation CD containing PDF files of all documentation.
- The distribution media (in tape cartridge form). Table 2–1 lists the contents of the distribution tape.

**Table 2–1. Distribution Table Contents**

| No. | File Name     | Description                            | DSORG | RECFM | LRECL | CYLS (3390) | DIR BLKS |
|-----|---------------|--|-------|-------|-------|-------------|----------|
| 1   | NEON.CNTL     | JCL library                            | PO    | FB    | 80    | 1           | 52       |
| 2   | NEON.ASM      | Assembler library                      | PO    | FB    | 80    | 1           | 2        |
| 3   | NEON.DBRMLIB  | Database Request Module (DBRM) library | PO    | FB    | 80    | 3           | 4        |
| 4   | NEON.EXEC     | SYSEXEC REXX library in VB format      | PO    | VB    | 255   | 15          | 74       |
| 5   | NEON.LIST     | Listings library                       | PO    | FBA   | 121   | 1           | 2        |
| 6   | NEON.LOAD     | Load library                           | PO    | U     | 0     | 48          | 114      |
| 7   | NEON.NEONMLIB | ISPF messages                          | PO    | FB    | 80    | 2           | 40       |
| 8   | NEON.NEONPLIB | ISPF panels                            | PO    | FB    | 80    | 7           | 410      |
| 9   | NEON.NEONTLIB | ISPF tables                            | PO    | FB    | 80    | 1           | 2        |
| 10  | NEON.TEXT     | Text library                           | PO    | VB    | 255   | 5           | 2        |
| 11  | NEON.SAMP     | Sample Web programs                    | PO    | FB    | 80    | 2           | 26       |



**Table 2–1. Distribution Table Contents (Continued)**

| No. | File Name     | Description                                  | DSORG | RECFM | LRECL | CYLS<br>(3390) | DIR<br>BLKS |
|-----|---------------|--|-------|-------|-------|----------------|-------------|
| 12  | NEON.EXECFB   | SSYSEXEC REXX library in FB format           | PO    | FB    | 80    | 17             | 76          |
| 13  | NEON.OBJ      | Object library                               | PO    | FB    | 80    | 20             | 42          |
| 14  | NEON.ATHEXEC  | SEF ATH library                              | PO    | FB    | 80    | 1              | 8           |
| 15  | NEON.EXCEXEC  | SEF EXC library                              | PO    | FB    | 80    | 1              | 6           |
| 16  | NEON.GLVEXEC  | SEF GLV library                              | PO    | FB    | 80    | 1              | 2           |
| 17  | NEON.RPCEXEC  | SEF RPC library                              | PO    | FB    | 80    | 1              | 2           |
| 18  | NEON.SQLEXEC  | SEF SQL library                              | PO    | FB    | 80    | 1              | 2           |
| 19  | NEON.TODEXEC  | SEF TOD library                              | PO    | FB    | 80    | 1              | 2           |
| 20  | NEON.TYPEXEC  | SEF TYP library for Shadow                   | PO    | FB    | 80    | 1              | 2           |
| 21  | NEON.ATHEXECW | Web Server SEF ATH library                   | PO    | FB    | 80    | 1              | 8           |
| 22  | NEON.EXCEXECW | Web Server SEF EXC library                   | PO    | FB    | 80    | 1              | 6           |
| 23  | NEON.GLVEXECW | Web Server SEF GLV library                   | PO    | FB    | 80    | 1              | 2           |
| 24  | NEON.TODEXECW | Web Server SEF TOD library                   | PO    | FB    | 80    | 1              | 2           |
| 25  | NEON.RPCLIB   | RPC load library                             | PO    | U     | 0     | 10             | 16          |
| 26  | NEON.TYPEXECW | SEF TYP library for Web Server               | PO    | FB    | 80    | 1              | 2           |
| 27  | NEON.WWWEXEC  | SEF master Web transaction library           | PO    | FB    | 80    | 1              | 6           |
| 28  | NEON.NEONEXEC | SEF NEON sample Web transaction library      | PO    | FB    | 80    | 1              | 22          |
| 29  | NEON.SAMPDATA | Sample HTML and GIF library                  | PO    | VB    | 255   | 8              | 184         |
| 30  | NEON.SWSCNTL  | Control applications Web transaction library | PO    | FB    | 80    | 1              | 52          |
| 31  | NEON.DATAM    | Sample maps for data mapping                 | PO    | FB    | 1024  | 1              | 2           |
| 32  | NEON.CMDEXEC  | Shadow SEF command ruleset                   | PO    | FB    | 80    | 1              | 4           |
| 33  | NEON.CMDEXECW | Shadow Web Server SEF command ruleset        | PO    | FB    | 80    | 1              | 2           |

**Table 2–1. Distribution Table Contents (Continued)**

| No. | File Name     | Description  | DSORG | RECFM | LRECL | CYLS<br>(3390) | DIR<br>BLKS |
|-----|---------------|--|-------|-------|-------|----------------|-------------|
| 34  | NEON.AHTML    | Shadow Web Server<br>Auto HTML                             | PO    | VB    | 19036 | 2              | 2           |
| 35  | NEON.CICSLOAD | CICS load library for<br>users of CICS-related<br>features | PO    | U     | 0     | 3              | 50          |
| 36  | NEON.CHGEXEC  | Shadow Event<br>Publisher sample<br>publication rule       | PO    | VB    | 1     | 1              | 1           |
| 37  | NEON.CHGEXECW | Shadow Event<br>Publisher sample<br>publication rule       | PO    | VB    | 1     | 1              | 1           |
| 38  | NEON.HTXLIB   | Shadow Event<br>Publisher sample<br>transformation rule    | PO    | VB    | 1     | 1              | 1           |

## Installation Features

The Shadow Mainframe Adapter Server product provides several features that are instrumental in the installation of the product. These include:

- The JCL required for installing the product.
- The INSTALL member for customizing your installation.
- The Shadow Mainframe Adapter Server load library.

## Product Installation Steps

The following steps are required for the installation of the Shadow Mainframe Adapter Server product::

- Step 1: Unload the CNTL Dataset.
- Step 2: Modify and Execute the Install Member.
- Step 3: APF-Authorize the Load Library.
- Step 4: Define new Slip Traps.
- Step 5: Create the Trace VSAM Datasets.
- Step 6: Set Up the Started Task JCL.
- Step 7: Define the Started Task Name to Your Security Product.
- Step 8: Provide VTAM Definitions.
- Step 9: Customize the Initialization EXEC.
- Step 10: Set Up the ISPF/SDF Dialogs.
- Step 11: Start the Shadow Mainframe Adapter Server.
- Step 12: Ensure that the Shadow Mainframe Adapter Client Has Been Installed.

## Step 1: Unload the CNTL Dataset.

The first library in the Shadow Mainframe Adapter Server distribution tape is the NEON.CNTL dataset. It contains the JCL needed for the rest of the installation process. To unload it, use the JCL (or equivalent) listed below:

```
//...          JOB
//UNLOAD       EXEC PGM=IEBCOPY
//TAPCNTL      DD DSN=NEON.CNTL,DISP=(OLD,PASS),
//              UNIT=TAPE,VOL=SER=xxxxxxx,
//              LABEL=(1,SL,EXPDT=98000)
//DSKCNTL      DD DSN=prefix.CNTL,DISP=(NEW,CATLG),
//              UNIT=SYSDA,VOL=SER=xxxxxxx,SPACE=(CYL,(10,1,50))
//SYSPRINT     DD SYSOUT=*
//SYSUT3       DD UNIT=SYSDA,SPACE=(CYL,1)
//SYSUT4       DD UNIT=SYSDA,SPACE=(CYL,1)
//SYSIN        DD *
COPY          INDD=((TAPCNTL,R)),OUTDD=DSKCNTL
//
```

### Notes:

**//DSKCNTL DD:** If you use SMS, substitute STORCLAS for UNIT on the //DSKCNTL DD statement and specify the proper storage class.

**VOL=SER=xxxxxxx:** The distribution tape you received contains a serial number in the format xxxxxxxx on its external label. Use this number in the JCL and in the INSTALL member in “Step 2: Modify and Execute the Install Member.” on page 2-5.

## Step 2: Modify and Execute the Install Member.

After you have unloaded the NEON.CNTL dataset, modify and execute the INSTALL member as follows:

1. Change the job card for your data center’s standards.
2. Change the TAPEVOL parameter to the volume serial number written on the Shadow Mainframe Adapter Server distribution tape.
3. Change the TAPEUNT parameter, if TAPE is not the correct unit name.
4. Change the DISKPFX parameter to the high-level dataset qualifier you are using for Shadow Mainframe Adapter Server libraries. The default is SDB.
5. Change the DISKUNT parameter if 3390 cannot be used to refer to the DASD unit on which Shadow Mainframe Adapter Server will reside.
6. Change the DISKVOL parameter to the volser of the DASD volume on which the Shadow Mainframe Adapter Server libraries will reside.
7. Submit INSTALL for execution. This member unloads the rest of the tape.

8. Check the output of the IEBCOPY step carefully. Checking the condition code may not be sufficient, since IEBCOPY can return the condition code zero even if nothing was copied.

### **Step 3: APF-Authorize the Load Library.**

The Shadow Mainframe Adapter Server load library must be APF-authorized. This can be done in one of the following ways:

- Put the load library in your LNKLST or LPALIB, and specify LNKAUTH=LNKLST in the IEASYSxx member of SYS1.PARMLIB for automatic authorization.
- Put the names of the load libraries and the volser of the disk on which they reside in SYS1.PARMLIB(IEAAPFxx). You must IPL to make the change effective. A site that does not want to IPL can use either an existing authorized library, the OS/390 or z/OS command, or any one of the major on-line OS/390 or z/OS performance/operations enhancement tools to add an entry for a new authorized library.



**Note:**

Ensure that the DB2 load library is ahead of the Shadow load library in LPALIB, LNKLST, or STEPLIB.

If you are running MVS/ESA version 4.3 or above, you can dynamically APF-authorize the Shadow load library by defining it in the PROGxx member of SYS1.PARMLIB and then issuing SET PROG=xx from the OS/390 or z/OS console. Use the following syntax in the PROGxx member:

```
APF ADD DSNNAME(NEON.SV040800.LOAD) VOLUME(xxxxxxxx)
```

### **Step 4: Define new Slip Traps.**

The Shadow Mainframe Adapter Server provides the ability to monitor and limit resources for client connections. If resource limits set are exceeded, the client connections can be cancelled automatically by the Shadow Mainframe Adapter Server. The Shadow Mainframe Adapter Server issues User abends to cancel these connections and in some instances OS/390 or Z/OS may take an SVC dump for these abends. Slip Traps with an option of NODUMP should be defined to avoid unnecessary SVC dumps to be taken for these events. The following Slip Traps should be added to the system parmlib dataset member IEASLPxx:

```
SLIP SET,C=U0222, ID=U222, A=NODUMP, END  
SLIP SET,C=U0322, ID=U322, A=NODUMP, END  
SLIP SET,C=U0522, ID=U522, A=NODUMP, END
```

These are the same types of Slip Traps you may already have set up for Systems abend x22, whereas Shadow Mainframe Adapter Server will issue a User abend.

- The U0222 abend is issued whenever a connection is cancelled manually by an individual using the Shadow Mainframe Adapter Server KILL option

under the Remote User Display or when connections are cancelled at shutdown.


- The U0322 abend is issued whenever a resource limit, such as the FAILSQLCPUTIME or FAILCPUTIME parameters, is exceeded .
- The U0522 abend is issued whenever a connection exceeds the FAILWAITTIME parameter.

## **Step 5: Create the Trace VSAM Datasets.**

Shadow Mainframe Adapter Server tracks communication and SQL processing events and records this information in a trace VSAM dataset. You can view this information using the trace browse application. The optional Shadow Event Facility™ (SEF), a component of Shadow , supports global variables and keeps this information in the SYSCHKx datasets.

### **Creating the Trace VSAM Datasets**

To create the trace VSAM datasets:

1. Change the dataset names, as appropriate.
    - Change NEON.SV040800.SDBB.TRACE to the name you use for the trace dataset. This name must be defined to the SDBTRACE ddname in the started task JCL.
    - If you are using SEF, change the NEON.SV040800.SDBB.SYSCHKx names to the names you will use for global variable support.
-  **Note:**  
SDBB should be the same as the Shadow subsystem name.
- When an installation has many thousands of data maps, a caching facility may be configured to allow for faster server restarts. Change NEON.SV040800.SDBB.DATAMAP.CACHE to the name you will use for the data map cache file.
  2. Ensure that the dataset size is large enough to contain the number of messages specified.
    - The CYLINDER parameter for the trace dataset should contain space for the number of messages that you have specified in the Shadow Mainframe Adapter Server BROWSEMAX parameter. Exactly 720 messages fit in a 3390 cylinder, and exactly 600 messages fit in a 3380 cylinder (each message is 1024 bytes).
    - The global variable checkpoint datasets must each be large enough to hold the number of variables specified in the Shadow Mainframe Adapter Server GLOBALMAX parameter. GLOBALMAX defaults to 5000 variables. Roughly 1180 variables fit in a 3380 cylinder.

- The data map cache file must be large enough to hold all data map binary images. We recommend that you express allocations for this cluster in Megabyte units. Each data map varies in size, depending on the number of individual fields defined within the map. The size, in bytes, needed to store each map is equal to the number-of-defined-fields, plus 1, times 1024.

▷ **Note:**

Each map will take up a minimum of 2K of storage, with an additional 1K for each column in the map. The best way to determine the estimated required storage is to multiply the average number of columns by the number of maps. To allow room for new maps, the user should increase the resulting number accordingly.

To determine the number of columns in a map, browse the map dataset via ISPF. The size parameter in the ISPF Browse display will give you the number of columns in a map so you can determine the size of the Data Map Cache dataset. The default of 10 megabytes for primary and 10 for secondary should be sufficient sizes for most sites.

3. Ensure that the VOL parameter specifies the volume serial number on which the dataset will reside.
4. Ensure that the data component name (the last dataset name) is the same as the cluster name, with an additional qualifier of "DATA".
5. Execute the **DEFINE CLUSTER** for the dataset. The easiest way to do this is to execute the DEFDIV member like a CLIST. For example, the following command could be used, assuming that the upper level qualifier for the CNTL dataset is NEON:

```
EX 'NEON.SV040800.CNTL(DEFDIV)'
```

As an alternative, you can include the **DEFINE CLUSTER** command in an IDCAMS step of a batch job.

## **Step 6: Set Up the Started Task JCL.**

The SDBB member of the CNTL library contains the JCL procedure needed to run the Shadow Mainframe Adapter Server main address space (started task). You must place the SDBB member in a procedure library that will be searched by the OS/390 or z/OS **START** command. This can be SYS1.PROCLIB, but does not have to be.

**Note:**

The restriction in Shadow Mainframe Adapter Server subsystem names is SDBx, where x is any alphanumeric character (A-Z and 0-9). The subsystem name is specified via the SSID parameter in the started task JCL.

## Customizing the Started Task JCL

Use the following steps to customize the JCL found in the SDBB member:

1. Add the name of the DB2 library to the DB2LIB, ensuring that the DB2 library is added ahead of the Shadow load library. Optionally, if the Shadow load library or DB2 load library has been placed in the linklist, you can remove that library from the STEPLIB concatenation and remove the parameter from the JCL.
2. If you plan to use IMS, add the name of the IMS RESLIB dataset to the IMSLIB parameter, and uncomment the parameter and the SDBRPCLB definition.
3. Add the high-level qualifier name of the Shadow Mainframe Adapter Server libraries to the HLQ parameter. This should properly set the Shadow Mainframe Adapter Server dataset allocations to their correct dataset names.
4. The SYSEXEC ddname must point to the Shadow REXX library. If your system's REXX and CLIST libraries are RECFM=FB, use the NEON.EXECFB dataset. If they are RECFM=VB, use the NEON.EXEC dataset. This dataset *must* contain the SDBx.IN00 initialization EXEC that will be modified in “Step 9: Customize the Initialization EXEC.” on page 2-14.
5. **(Perform this step if you wish to run the sample VSAM RPC IVP)** Modify and execute the member DEFSTAFF in the NEON.SV040800.CNTL dataset. DEFSTAFF will allocate and populate the sample VSAM dataset. Uncomment the SDBVS01 ddname in order to allocate the VSAM dataset to the Shadow Mainframe Adapter Server.

**Note:**

We recommend that you disable ABEND-AID when using Shadow products. To disable ABEND-AID, set ddname //SYSABEND in SDBB to DUMMY. The EOT processing is faster with ABEND-AID disabled.

6. **(Perform this step only if you are installing the Data Mapping Facility)** Use either the supplied datasets or create your own, as follows:
  - If you use the supplied datasets, remove the comment mark. The supplied dataset name is SDBMAPP. For more information, see “Using Supplied Datasets” on page 2-10.

- If you create your own datasets, they *must* be referenced in the start-up JCL by concatenating your map dataset names below the default name. If you do not add your map names, the mapping facility will continue to use the default. For more information, see “Creating Datasets” on page 2-10.
- If you will use the data mapping cache facility to expedite server restart (necessary only when your installation has many thousands of data maps), uncomment the DD statement for SDBMAPL in the sample JCL. Change the dataset name to match the name used when defining the data mapping cache linear datasets.

### **Using Supplied Datasets**

To use the supplied datasets, remove the comment mark from the line shown in the following excerpt from the start-up JCL:

```
//*****
//*      THE FOLLOWING DDNAME MAY BE UNCOMMENTED IF DATA MAPS ARE      *
//*      TO BE USED IN A PRODUCTION ENVIRONMENT.                        *
//*****
//*SDBMAPP      DD   DISP=SHR,DSN=&HLQ..DATA.MAPS                        *
//*****
```

### **Creating Datasets**

If you create your own map dataset, it must have the following attributes:

- For FB (fixed block):
  - Record Format is FB.
  - Record Length is 1024.
  - Block Length is 20480.
  - Dataset Organization is PO.
- For VB (variable block):
  - Record Format is VB.
  - Record Length is 19036.
  - Block Length is 19040.
  - Dataset Organization is PO.

## **Step 7: Define the Started Task Name to Your Security Product.**

If you are running a security product such as RACF, ACF2, or Top Secret, you may have to define userids for the Shadow Mainframe Adapter Server address space and set up access rules so that Shadow Mainframe Adapter Server can use the datasets it needs.

Table 2–2 summarizes the access requirements for Shadow Mainframe Adapter Server, as distributed. If you customize the dataset names for your installation, be sure to give those dataset names the appropriate access.



As noted earlier, all datasets can be shared between different Shadow Servers with the exception of trace and SYSCHKx datasets. These datasets must be unique to each copy of Shadow Mainframe Adapter Server. This is true whether or not the two Shadow Servers are on the same machine.

**Table 2–2. Access Requirements for Shadow Mainframe Adapter Server**

| Dataset Name   | Access        |
|--|---------------|
| IBM.DB2LIB   | EXECUTE       |
| NEON.SV040800.LOAD                                   | EXECUTE       |
| NEON.SV040800.EXEC                                   | READ          |
| NEON.SV040800.SDBB and SDBB.TRACE                    | READ, WRITE   |
| NEON.SV040800.SDBB and SDBB.SYSCHK1 and SDBB.SYSCHK2 | READ, WRITE   |
| NEON.SV040800.SDBB and SDBB.*.EXEC                   | READ, WRITE   |
| NEON.SV040800.RPCLIB                                 | READ, EXECUTE |
| QUICKREF.LINKLIB <sup>a</sup>                        | EXECUTE       |
| QUICKREF.DATABASE*                                   | READ          |

- a Sites that have MVS/Quick-Ref installed should provide access to the QUICKREF.LINKLIB and QUICKREF.DATABASE datasets. If MVS/Quick-Ref is not installed, ignore these two datasets.



**Note:**

Running Shadow without giving its address space enough authorization to access its own datasets is one of the most common installation problems.

## Defining the Started Task to RACF

An example of defining the name of Shadow (SDBB) started task to RACF follows:



**Note:**

If you run into any problems performing this step, please see your RACF administrator.

```
ADDUSER SDBB NAME('TEST SHADOW SDBB') DFLTGRP(SYS1) -
  OMVS(UID(nnnn) HOME('/')) OWNER(<your profile owner>)
LISTUSER SDBB OMVS
```

```
RDEFINE STARTED SDBB.** -
  LEVEL(0) OWNER(<your profile owner>) STDATA(USER(SDBB)
  GROUP(SYS1))
```

```
SETROPTS RACLIST(STARTED) REFRESH  
  
PERMIT SDBB.** CLASS(STARTED) ID(SDBB) ACCESS(ALTER) GENERIC  
SETROPTS RACLIST(STARTED) REFRESH  
RLIST STARTED SDBB.** STDATA AUTHUSER
```

**Note:**

The UID does not have to be 0 and default group does not have to be SYS1. If you see the userid for the started task shows +'s, something failed during the RACF definition.

If you choose a port number for Shadow that is less than or equal to 1024, you must use a UID equal to 0. This is not recommended since UID=0 has special privileges.

## Defining the Started Task to CA-Top Secret

To properly define Shadow to CA-Top Secret:

1. Set up the facility entry for Shadow Mainframe Adapter Server with the following options (most are CA-Top Secret defaults):

```
AC ( USERx=NAME=SHADOWT )  
FAC ( SHADOWT=ACTIVE , SHRPRF , ASUBM , NOABEND , SUAS , NOXDEF )  
FAC ( SHADOWT=PGM=SDB2IN , ID=ST , LUMSG , STMSG , SIGN ( M ) ,  
INSTDATA , RNDPW )  
FAC ( SHADOWT=NOPROMPT , NOAUDIT , RES , WARNPW , NOTSOC , LCFTRANS )  
FAC ( SHADOWT=MSGLC , NOTRACE , NODORMPW , NONPWR , NIIMSXTND )  
FAC ( SHADOWT=MODE=FAIL , LOG ( INIT , MSG , SEC9 , SMF )
```

2. Add the master facility SHADOWT to the ACID for the Shadow started task as follows:

```
TSS ADDTO(SDBB) MASTFAC(SHADOWT)
```

Access to the Shadow started task can be handled by the options defined for the SHADOWT facility. All users who sign onto Shadow will need to be authorized through FACILITY(SHADOWT).

## Step 8: Provide VTAM Definitions.

If you are using SNA communications with Shadow Mainframe Adapter Server commands, you must provide the following definitions for VTAM:

- APPL statements that define the Shadow Mainframe Adapter Server application.
- Cross Domain Resource Members (CDRMs) that define inter-domain connections.
- Mode table entries that determine certain communication parameters.

## Coding the VTAM APPL Statement

In each system, only one APPL statement is required for Shadow Mainframe Adapter Server. You should code the statement as follows:

```
SDBAPPL VBUILD TYPE=APPL
applid      APPL AUTH=(ACQ) ,           <--DEFINE THE MAJOR NODE
            APPC=YES ,                 <--AUTHORIZE USE OF LU 6.2
            SECACPT=AVPV ,
            DSESLIM=20
            DMINWNL=10
            DMINWNR=10
            MODETAB=SDB2MODE
```

Where:

- **applid** specifies the VTAM application name for the Shadow Mainframe Adapter Server system. Since this name must be unique in your network, it is common to make some CPU-specific identifier, such as the SMFID, part of the netname. This name will match the APPLID operand of the **DEFINE LINK** command.
- **AUTH=(ACQ)** permits Shadow Mainframe Adapter Server to issue the OPNDST macro. The OPNDST macro allows Shadow Mainframe Adapter Server to acquire sessions with other Shadow Servers running on different systems.
- **APPC=YES** permits Shadow Mainframe Adapter Server to use APPCCMD (LU 6.2) macros to communicate with other Shadow Mainframe Adapter Server systems.
- **SECACPT=CONV** allows certain security information to be accepted by Shadow Mainframe Adapter Server. This must be coded exactly as specified for APPC sessions to be activated properly.
- **DSESLIM=20** sets the defined session limit for this system at 20. This must be coded exactly as specified for APPC sessions to be activated properly.
- **DMINWNL=10** sets the defined minimum number of contention winner sessions for this local system at 10. This must be coded exactly as specified for APPC sessions to be activated properly.
- **DMINWNR=10** sets the defined minimum number of contention loser sessions for this local system at 10. This must be coded exactly as specified for APPC sessions to be activated properly.
- **MODETAB=modetab** designates the name of a VTAM LOGMODE table that contains an LU 6.2 mode table entry (MODEENT). The format of this table entry is discussed in “Defining the LU 6.2 VTAM Mode Table Entry” on page 2-14. This parameter must be supplied, and it must contain a valid LU 6.2 mode entry for APPC sessions to be activated properly.

- ▶ **Note:**  
A sample APPLID definition is provided in the SDBAPPL member of NEON.SV040800.CNTL.

## Defining the LU 6.2 VTAM Mode Table Entry

To create an LU 6.2 mode table entry, find an existing VTAM mode table that contains the LU 6.2 session parameters. If you cannot find an existing entry, create one similar to the following sample mode table entry:

```
SDBMODE MODETAB
SDB2MODE MODEENT LOGMODE=SDB2MODE,
      FMPROF=X'13',TSPROF=X'07',PRIPROT=X'B0',
      SECPROT=X'B0',COMPROT=X'50B1',RUSIZES=X'8888',
      PSNDPAC=X'05',SRCVPAC=X'05',SSNDPAC=X'05',
      PSERVIC=X'0602000000000000000000000000300',TYPE=X'00'
MODEEND
```

- ▶ **Note:**  
The source for this sample can be found in the SDBMODE member of NEON.SV040800.CNTL.

## Step 9: Customize the Initialization EXEC.

The initialization EXEC is a REXX program used to set product parameters and define links and databases. The name of the initialization EXEC must be SDBxIN00, where x is the last character of the 4-character subsystem ID. Both variable blocked (.EXEC) and fixed blocked (.EXECFB) datasets have been provided.

- ▶ **Note:**  
Since the default subsystem ID is SDBB, the EXEC is generally named SDBBIN00.

The EXEC must be placed in the library that is allocated to the SYSEXEC ddname in the SDBB started task procedure. It is recommended that you place the SDBxIN00 member in a separate dataset so that future maintenance will not overwrite your modified member.

- ▶ **Note:**  
The initialization EXEC *must be completed in all uppercase characters*. The only exception is with certain operand values, which can be coded in lowercase if the actual operand is lowercase. Do not code a lowercase operand value if the actual value is uppercase.

## Sample EXECs

When properly modified, the sample initialization EXEC, SDBBIN00, in the NEON.EXEC REXX library will initialize Shadow for client-server processing.

The sample initialization EXEC that is shipped in member SDBBIN00 of the NEON.EXEC(FB) dataset is set up so that features can be turned on and off by simply modifying an IF statement. For example, to turn on LU 6.2 support, simply modify the following statement:

```
IF 1=2 THEN /* LU 6.2 CLIENT/SERVER? */
"MODIFY PARM NAME (APPLID) VALUE(SDBIP00)"
```

To read as follows:

```
IF 1=1 THEN /* LU 6.2 CLIENT/SERVER? */
"MODIFY PARM NAME (APPLID) VALUE(SDBIP00)"
```

Specify the APPLID for the value.

## Tailoring the Initialization EXEC Structure

The initialization EXECs can be as simple or as complex as you want. However, there are a few general guidelines you should follow:

- Step A: Setting Up General Started Task Parameters
- Step B: Enabling REUSETHEADS
- Step C: Inputting the License Code
- Step D: Defining the Shadow ISPF Dialog Datasets
- Step E: Issuing a DEFINE RULESET statement for each SEF Ruleset
- Step F: Defining Network Connectivity
- Step G: Setting the Local VTAM APPLID Value

## Step A: Set Up General Started Task Parameters.

Set up the following general product parameters:

```
"MODIFY PARM NAME(BROWSEMAX)           VALUE(100000)"
"MODIFY PARM NAME(ODBCCATALOGLEVEL)     VALUE(3)"
"MODIFY PARM NAME(FAILSQLCPUTIME)       VALUE(120)"
"MODIFY PARM NAME(AUTOSTATICSQL)        VALUE(NO)"
"MODIFY PARM NAME(TRACEAUTHEVENTS)      VALUE(NO)"
"MODIFY PARM NAME(SHARESUBPOOLZERO)     VALUE(NO)"
"MODIFY PARM NAME(ACF2SAFCALL)          VALUE(YES)"
"MODIFY PARM NAME(DEFAULTTDB2SUBSYS)    VALUE(NONE)"
"MODIFY PARM NAME(USERABENDKILL)        VALUE(YES)"
"MODIFY PARM NAME(ROLLBACKRPCABEND)     VALUE(YES)"
"MODIFY PARM NAME(DB2CONCURRENTMX)      VALUE(500)"
"MODIFY PARM NAME(WLMCONNECT)           VALUE(COMPAT)"
```

**Note:**

Refer to the *Shadow Started Task Parameter Guide* for more information about these parameters.

The BROWSEMAX parameter has the default value of 100000. Based on this value, the dataset size for the trace browse VSAM file can be calculated by figuring 1K per line. Changing the value of this parameter in the Shadow initialization EXEC will cause the trace browse to be reformatted at the next startup, with a consequential loss of all pre-existing data.

Also, make sure that the value of the BROWSEMAX parameter is set in accordance with the value you set for Trace Browse. See “Step 5: Create the Trace VSAM Datasets.” on page 2-7 of this chapter.

It is recommended that you accept the default value of the remaining parameters during Initialization. There may be instances in which you will later need to change some of these values, but these instances will be noted throughout the Shadow Documentation.

## Step B: Enable REUSETHEADS

The REUSETHEADS parameter controls whether or not threads should be reused. If this flag is set to YES, each thread will be reused a number of times if possible. If this flag is set to NO, a new thread will always be created for each new inbound session. Thread reuse may reduce CPU resource utilization quite considerably when DB2 threads are used frequently and/or client userids are cached and reused for persistent session support.

You can enable REUSETHEADS, the server side reusable connections, by setting the following started task parameters:

```
"MODIFY PARM NAME(REUSETHEADS) VALUE(YES)"
"MODIFY PARM NAME(TARGETTHREADCOUNT) VALUE(500)"
"MODIFY PARM NAME(THREADTIMEOUT) VALUE(300)"
```

Where:

- **REUSETHEADS** controls whether threads should be reused or not. Possible values are:
  - **YES:** Each thread will be reused a number of times if possible.
  - **NO:** (Default) A new thread will always be created for each new inbound session.
- **TARGETTHREADCOUNT** controls the target number of threads in some UDP and TCP execution modes. The value controls the number of

subtasks created during product startup to handle inbound UDP datagrams and TCP sessions.

- **THREADTIMEOUT** controls how long a thread will wait for new work to be assigned to it. When the time limit is reached the thread terminates. Setting too small a value will cause thread churning. Setting too high a value may leave too many idle threads.

## Step C: Input the License Code.

You must specify your license code using the following line:

```
"MODIFY PARM NAME(LICENSECODE) VALUE(licensecodestringval)"
```

Replace the string `licensecodestringval` with your personal license code that you received via your confirmation e-mail. This code includes (in encrypted format) the product name and features available for the site, the CPU on which the product is licensed to run, and the duration of the license period.

Since the code is encrypted, you must enter the value exactly as you receive it. Failure to do so will result in the product's inability to start. You can see the decrypted form of your license code after you have installed Shadow Mainframe Adapter Server on your machine.



### **Note:**

Shadow provides the ability to continue operation during disaster recovery or a disaster recovery test on an unlicensed CPU. For more information about the requirements.

## Step D: Define the Shadow ISPF Dialog Datasets

To access the Shadow ISPF panels without having to manually allocate them to a TSO user's logon proc or allocations, you can optionally define the ISPF datasets in the initialization EXEC. This will make them accessible by anyone invoking the Shadow/REXX command as long as the Shadow started task is active.

To define the ISPF datasets in the initialization EXEC, set the following parameters:

```
"MODIFY PARM NAME(EXECDSNAME) VALUE(NEON.SV040800.EXEC)"
"MODIFY PARM NAME(ISPLLIBDSNAME) VALUE(NEON.SV040800.LOAD)"
"MODIFY PARM NAME(ISPMLIBDSNAME) VALUE(NEON.SV040800.NEONMLIB)"
"MODIFY PARM NAME(ISPPLIBDSNAME) VALUE(NEON.SV040800.NEONPLIB)"
"MODIFY PARM NAME(ISPTLIBDSNAME) VALUE(NEON.SV040800.NEONTLIB)"
```

Where:

- **EXECDSNAME** is the compiled REXX EXEC dataset name.
- **ISPLLIBDSNAME** is the ISPLLIB dataset name.
- **ISPMLIBDSNAME** is the ISPMLIB dataset name.
- **ISPPLIBDSNAME** is the ISPPLIB dataset name.

- **ISPTLIBDSNAME** is the ISPTLIB dataset name.

Even though the Shadow load library is allocated to Shadow Mainframe Adapter Server, it is still required to use an ISPF LIBDEF for the Shadow load library before invoking the Shadow/REXX EXEC that brings up the ISPF/SDF dialogs (see “Step 10: Set Up the ISPF/SDF Dialogs.” on page 2-23). This can be avoided by copying the Shadow load modules to a linklist dataset or to a dataset allocated to the user’s ISPLLIB allocation.

## Step E: Issue a DEFINE RULESET statement for each SEF Ruleset.

Issue a DEFINE RULESET statement for each SEF ruleset. During initial installation, you must change each DSNAME operand to match the dataset names actually installed with the product.

```

"MODIFY PARM NAME(SEFV3COMPATIBLE) VALUE(NO)"
"DEFINE RULESET NAME(ATH)"
    "RULETYPE(ATH)"
    "DSNAME('NEON.SV040800.SDBB.ATH.EXEC')"
```

```

"DEFINE RULESET NAME(CHG)"
    "RULETYPE(CHG)"
    "DSNAME('NEON.SV040800.SDBB.CHG.EXEC')"
```

```

"DEFINE RULESET NAME(CMD)"
    "RULETYPE(CMD)"
    "DSNAME('NEON.SV040800.SDBB.CMD.EXEC')"
```

```

"DEFINE RULESET NAME(EXC)"
    "RULETYPE(EXC)"
    "DSNAME('NEON.SV040800.SDBB.EXC.EXEC')"
```

```

"DEFINE RULESET NAME(GLV)"
    "RULETYPE(GLV)"
    "DSNAME('NEON.SV040800.SDBB.GLV.EXEC')"
```

```

"DEFINE RULESET NAME(RPC)"
    "RULETYPE(RPC)"
    "DSNAME('NEON.SV040800.SDBB.RPC.EXEC')"
```

```

"DEFINE RULESET NAME(SQL)"
    "RULETYPE(SQL)"
    "DSNAME('NEON.SV040800.SDBB.SQL.EXEC')"
```

```

"DEFINE RULESET NAME(TOD)"
    "RULETYPE(TOD)"
    "DSNAME('NEON.SV040800.SDBB.TOD.EXEC')"
```

```

"DEFINE RULESET NAME(TYP)"
    "RULETYPE(TYP)"
    "DSNAME('NEON.SV040800.SDBB.TYP.EXEC')"
```

END

## Step F: Define Network Connectivity.

Shadow can support the following types of network connectivity:

- LU 6.2
- TCP/IP



## Defining LU 6.2 Connectivity

For the Shadow Mainframe Adapter Client code to successfully communicate with Shadow Mainframe Adapter Server via LU 6.2, the following MODIFY PARM command must be issued to define the VTAM APPLID. The syntax is as follows:

```
"MODIFY PARM NAME(APPLID) VALUE(value)"
```

Where:

- **APPLID** specifies the name of the parameter to be modified. In this case, the value APPLID must be specified to indicate that the local VTAM APPLID is being specified.



### Note:

You must define to VTAM the VTAM application IDs used by Shadow Mainframe Adapter Server before starting Shadow Mainframe Adapter Server. VTAM APPLIDs are defined in members of the SDB.VTAMLST dataset. For more information, see “Step 8: Provide VTAM Definitions.” on page 2-12.

## Defining TCP/IP Support

If you are configuring Shadow Mainframe Adapter Server to use TCP/IP, you must define a TCP/IP port number in the Shadow initialization EXEC. Optionally, you can reserve this port number within the TCP/IP stacks profile dataset so that other tasks cannot access this port and it becomes exclusive for the indicated Shadow Mainframe Adapter Server.

- The IBM TCP/IP port number is defined in the TCPIP.PROFILE dataset. This dataset is pointed to by the PROFILE ddname in the TCP/IP started task.
- The Interlink port number is defined in the INTC.PARM(DNRSVC00) dataset. This dataset is pointed to by the SYSPARM ddname in the INTERLINK started task.



### Note:

The IBM TCP/IP port number can be defined either in the TCPIP.DATA(PROFILE) member or the TCP/IP.PROFILE dataset, depending on the version of OS/390 or the z/OS you are using.

## Defining TCP/IP OE Sockets Support

Set the port numbers for OE Socket Support. These values are only used if you are running OE sockets. OE sockets can run over TCP/IP, MVS TCP/IP, and other TCP/IP implementations.

To define TCP/IP OE Sockets support:

1. Ensure that the Shadow Mainframe Adapter Server started task ID has been defined to OMVS. The RACF command is as follows:

```
ALTUSER SDBB OMVS(UID(x))
```

Where *x* is the UID, which specifies the user identifier between 0 and 2 147 483 647. Shadow Mainframe Adapter Server does NOT require superuser status (0) unless you assign the port number to be 1024 or less.



**Note:**

If you run into any problems performing this step, please see your RACF administrator.

2. Assign port numbers to OE Sockets TCP/IP, using the **MODIFY PARM** command. Port numbers are assigned as follows:

```
"MODIFY PARM NAME(OEPORTNUMBER) VALUE(1200)"
"MODIFY PARM NAME(IBMPORTNUMBER) VALUE(0000)"
"MODIFY PARM NAME(TRACEOERW) VALUE(YES)"
"MODIFY PARM NAME(OEKEEPALIVETIME) VALUE(30)"
"MODIFY PARM NAME(OESTACK) VALUE(TCPIP)"
```

Where:



**Note:**

The IBM port number parameter must have a value of 0000.

- **OEPORTNUMBER** sets the port number used to LISTEN for, and ACCEPT all inbound OE Sockets TCP/IP sessions. This port number should be reserved for exclusive use by the main product address space. Each copy of the main product address space will need its own separate port number if TCP/IP is being used. There is a default value for this port number if it is not set in the initialization EXEC. The port number can be set to a string of "ANY". This is a special value used to show that the system should assign an ephemeral port number for use by the product.
- **IBMPORTNUMBER** sets the port number used to LISTEN for, and ACCEPT all inbound TCP/IP sessions. This port number should be reserved for exclusive use by the main product address space. Each copy of the main product address space will need its own separate port number if TCP/IP is being used. There is a default value for this port number if it is not set in the initialization EXEC. The port number can be set to a string of "ANY". This is a special value used to show that the system should assign an ephemeral port number for use by the product.

- **TRACEOERW** controls whether or not IBM OE Sockets TCP/IP read/write events should be traced. Possible values are:
    - **YES:** (Default) IBM OE Sockets TCP/IP read/write events will be traced.
    - **NO:** IBM OE Sockets TCP/IP read/write events will not be traced.
  - **OEKEEPALIVETIME** utilizes the TCP/IP keepalive facility to detect that a connection is likely no longer valid and force a disconnect. If no data is transferred on a connection in the interval coded here, then the connection is tested and if no response is received, it is disconnected and any resources using it are freed. The smaller the value, the sooner invalid connections will be cleaned up but the possibility of disconnecting slow connections will be greater.
  - **OESTACK** specifies the name of the OE TCP/IP stack that should be used. For OE TCP/IP, this parameter is optional. If this parameter is not set, then the default OI stack will be used. If this parameter is used to select an OE TCP/IP stack, then the value must be one of the SUBFILESYSTYPE values specified in the PBXPRMxx PARMLIB member.
3. **(Optional)** Define values for SSL port numbers if SSL sessions are being used. Set the following parameter, using the **MODIFY PARM** command:

```
"MODIFY PARM NAME(OESSLPORTNUMBER) VALUE(1300)"
```

Where:

- **OESSLPORTNUMBER** controls whether or not IBM TCP/IP read/write events should be traced or not. Possible values are:
  - **YES:** IBM TCP/IP read/write events will be traced.
  - **NO:** (Default) IBM TCP/IP read/write events will not be traced.



**Note:**

SSL port numbers should only be set if SSL sessions are being used. The SSL port number must not be the same as the non-SSL port number.

4. If you are running multiple IBM TCP/IP OE stacks, and you want this Shadow Mainframe Adapter Server to use a stack other than the default stack, you must specify the other stack via the following parameter:

```
"MODIFY PARM NAME (OESTACK) VALUE (XXXX)"
```

Where:

- **OESTACK** specifies the name of the OE TCP/IP stack that should be used. For OE TCP/IP, this parameter is optional. If this parameter is not set,

then the default OI stack will be used. If this parameter is used to select an OE TCP/IP stack, then the value must be one of the SUBFILESYSTYPE values specified in the PBXPRMxx PARMLIB member. The value XXXX is one of the SUBFILESYSTYPE values specified in the BPXPRMxx PARMLIB member.

5. Modify your SYS1.PARMLIB(BPXPRMxx) member that configures Open Edition and ensure that the following parameters are set:
  - **MAXFILEPROC** specifies the maximum number of file descriptors that a single user is allowed to have concurrently active or allocated. You should set this parameter to the maximum number of users you plan to have connected to a single Shadow Mainframe Adapter Server.
  - **MAXSOCKETS** controls the maximum number of sockets per address space and needs to be raised to the maximum number of users you plan to have connected to a single Shadow Mainframe Adapter Server. This parameter is located under the FILESYSTYPE TYPE(INIT) definition.
  - **MAXPROCSYS** specifies the maximum number of processes that OS/390 UNIX will allow to be active concurrently. This parameter should be set to the maximum number of Shadow Mainframe Adapter Server connections you expect to have open at one time for all your Shadow Servers combined.

### ***Defining Interlink's TCP/IP Support***

1. Set the subsystem name and port number for Interlink TCP/IP. These values should only be set if you are running the Interlink version of TCP/IP on the host.

```
"MODIFY PARM NAME( ITCSSUBSYSTEM) VALUE(ACSS) "  
"MODIFY PARM NAME( ITCPORTNUMBER) VALUE(1200) "  
"MODIFY PARM NAME( TRACEITCIPRW) VALUE(YES) "
```

Where:

- **ITCSUBSYSTEM** specifies the local ITC/IP subsystem name.
  - **ITCPORTNUMBER** specifies the interlink TCP/IP port number.
  - **TRACEITCIPRW** specifies the trace ITC/IP read/write events.
2. **(Optional)** Define SSL port number values:

```
"MODIFY PARM NAME( ITCSSLPORTNUMBER) VALUE(1300) "
```

Where:

- **ITCSSLPORTNUMBER** sets the port number used to LISTEN for, and ACCEPT all inbound encrypted Interlink TCP/IP sessions. This port number should be reserved for use only by the main product address space. Each copy of the main product address space will need its own port number if SSL over Interlink is being used. There is a default value for the SSL port number if the value is not set in the initialization EXEC.

**Note:**

SSL port numbers should only be set if SSL sessions are being used. The SSL port number must not be the same as the non-SSL port number.

**Step G: Set the Local VTAM APPLID Value.**

You only need to set this value if you are using LU 6.2 to connect clients to the host.

```
"MODIFY PARM NAME(APPLID) VALUE(SDBIP00)"
"MODIFY PARM NAME(TRACELU62RDWR) VALUE(YES)"
```

Where:

- **APPLID** specifies the VTAM application ID.
- **TRACELU62RDWR** specifies the trace LLU 6.2 read/write events.

**Step 10: Set Up the ISPF/SDF Dialogs.**

To set up the ISPF/SDF application:

1. Edit the SHADOW member of NEON.EXEC and change the parameter LLIB as follows:
 

```
llib=<the Shadow load library>
```
2. Copy the SHADOW member of NEON.EXEC to a dataset allocated to all TSO users' SYSPROC allocation.

If the Shadow ISPF datasets were defined in the Shadow initialization EXEC, all of the required ISPF/SDF dataset allocations are allocated dynamically once the Shadow initialization EXEC is invoked, as long as the Shadow Mainframe Adapter Server is up and running.

**Note:**

If you are using TSO Command Limiting, a feature of ACF2 that requires access permissions to execute TSO commands, you must define all the Shadow ISPF TSO commands to your security product before you can use the Shadow ISPF/SDF dialogs. Failure to do so will result in **SDB** command "not found" error messages when attempting to execute the ISPF/SDF dialogs.

**Step 11: Start the Shadow Mainframe Adapter Server.**

From an operational perspective, Shadow Mainframe Adapter Server is an OS/390 or z/OS started task. It can be started with the **START** command and stopped with the **STOP** command. In normal circumstances, Shadow Mainframe

Adapter Server will be started at system start-up (IPL) and stopped just before the system is shut down. In other words, it is designed for continuous operation.

## Starting Shadow Mainframe Adapter Server

To start Shadow Mainframe Adapter Server, use the OS/390 or z/OS **START** command as follows:

```
S SDBB
```

If you are using an automation package to start your system, you should “hang” the **START** command for Shadow Mainframe Adapter Server off of the VTAM initialization complete message (IST020I), the TCP/IP initialization complete message (EZB6473I), or the DB2 initialization complete message (DSN9022I).

## Stopping Shadow Mainframe Adapter Server

To stop Shadow Mainframe Adapter Server, use the OS/390 or z/OS **STOP** command as follows:

```
P SDBB
```

Shadow Mainframe Adapter Server will wait for all active conversations to end before terminating, so it may take a while to shut down. If you cannot wait for Shadow Mainframe Adapter Server to terminate normally, use the **CANCEL** command as follows:

```
CANCEL SDBB
```

When you cancel Shadow Mainframe Adapter Server, all active conversations are terminated with an abend, and the product should shut down immediately.

## ***Step 12: Ensure that the Shadow Mainframe Adapter Client Has Been Installed.***

See the *Shadow Mainframe Adapter Client for Natural Client Installation and Administration Guide*.

# Shadow Mainframe Adapter Server: Authorizing Access To Resources

---

---

This chapter covers the steps for authorizing access to Shadow Mainframe Adapter Server resources. Shadow Mainframe Adapter Server is the server component of the Shadow product.

Topics include the following:

- Overview
- Protected Resources
- How Resource Access Is Determined
- Defining Shadow Resources to RACF
- Defining Shadow Resources to CA-Top Secret
- Defining Shadow Resources to ACF2
- Defining Shadow ISPF Load Modules
- Using the RACF Pass Ticket
- Note on Started Task Security
- Controlling Information Access with the TRACEDATA Resource
- Resource Security for Test Versions of Shadow Mainframe Adapter Server

## Overview

Shadow Mainframe Adapter Server provides protection for its resources using RACF classes, CA-Top Secret classes, and ACF2 generalized resource rules.

**Note:**

You need not authorize access to resources if you are running a trial version of Shadow.

The overall RACF class (or resource type, for ACF2) for Shadow Mainframe Adapter Server is specified with the Shadow Mainframe Adapter Server `RESOURCETYPE` parameter found in the `SDBxIN00` initialization EXEC. If not explicitly specified, `RESOURCETYPE` defaults to `NON`. This value disables all product authorization checking.

If you choose to set this parameter to the subsystem name, `SDx`, where `x` is the fourth letter of the subsystem name (usually “B”), you will be able to run multiple copies of Shadow Mainframe Adapter Server and either share the authorization rules or keep them separate.

During initial installation of the product, it is recommended that you leave this parameter value set to `NON`, *if possible*. This is because during initial installation,

most sites install the Shadow Mainframe Adapter Server on test z/OS systems, to which access is already limited and which are not directly exposed to the Internet. You may want to avoid the complexity of defining security subsystem generalized resource rules during this stage of deployment.

If you elect to leave generalized resource checking disabled at this stage, a security exposure may exist. Anyone with a valid TSO userid can gain access to the Shadow Mainframe Adapter Server ISPF control application, where they will be fully authorized to perform any function provided by the interface. This assumes, however, that the user has sufficient information at hand to logon to TSO/E and then gain access to the ISPF/SDF application.

## Protected Resources

The resources (or entities, in RACF terminology) protected by the product security mechanism are shown in Table 3–1. The resource names are fixed and cannot be modified by the customer.

**Table 3–1. Resources Protected by the Product Security Mechanism**

| Resource Name        | Description  |
|----------------------|--|
| ACI.aci-mapname      | Access to an ACI (Advanced Communication Interface) service definition for Shadow Mainframe Adapter Client for Natural users.  |
| ADA.ADABAS-file-name | Access to an ADABAS file name. See the <i>Shadow Mainframe Adapter Client for ADABAS: Shadow Mainframe Adapter Server Administration and Shadow Interface for ADABAS Administration</i> guide.   |
| ADAxxxxx.FILyyyyy    | Access to ADABAS file ID number. See the <i>Shadow Mainframe Adapter Client for ADABAS: Shadow Mainframe Adapter Server Administration and Shadow Interface for ADABAS Administration</i> guide. |
| CICSCONNECTIONS      | Access to monitor and control CICS connections.  |
| CONTROLBLOCKS        | Shadow Mainframe Adapter Server internal data structures.  |
| DATABASES            | Databases that are defined to Shadow Mainframe Adapter Server.   |
| DATAMAP              | Access to the Data Mapping Facility.   |
| FILE                 | Shared files that are defined to Shadow Mainframe Adapter Server.  |
| FILETYPE             | Access to the server's file-suffix/MIME-type control table.  |
| GLOBALS              | Access to global variables.  |
| IMSLTERM             | Tables correlating userids or TCP/IP addresses to LTERM to legacy LTERM security can be supported using an APPC interface.   |
| LINKS                | Communication links that are defined to Shadow Mainframe Adapter Server.   |
| PARMS                | Access to the ISPF/SDF parameter display.  |
| RPC.<rpc_name>       | RPC-based security. Not applicable.  |
| SDB                  | Access to the ISPF/SDF interactive control facility.   |



**Table 3–1. Resources Protected by the Product Security Mechanism (Continued)**

| Resource Name | Description  |
|---------------|--|
| SEF           | Access to the Shadow Event Facility™ dialogs. Not applicable.  |
| TOKENS        | Access to the Shadow Mainframe Adapter Server tokens display.  |
| TRACEBROWSE   | Access to the trace browse facility.   |
| TRACEDATA     | Access to the following: <ul style="list-style-type: none"> <li>• SQL information.</li> <li>• An uncensored view of the wrap-around trace.</li> <li>• The underlying binary trace records.</li> </ul> See “Controlling Information Access with the TRACEDATA Resource” on page 3-10. |
| USERS         | Access to the attached/remote users applications.  |

## How Resource Access Is Determined

When you invoke one of Shadow Mainframe Adapter Server’s facilities, the combination of your userid and the facility’s class are passed to the security package for authorization checking. The security package will use the rules that you specify to determine whether access should be allowed.

To expedite future authorization checks of an identical request, Shadow Mainframe Adapter Server keeps the results of all security checks in protected storage.

The “look-aside” security check information is saved on a Task Control Block (TCB) basis and remains in effect until the TCB terminates. If you are initially denied access but later have your security profile changed to allow access, you must exit the ISPF/SDF application to terminate its TCB. Depending on the security package, you may have to take other actions. Under ACF2, for example, you must issue the **ACFRESET** command. All security authorization events are logged in the trace browse facility, and if access is denied, a message is produced.

The type of access you request—ADD/ALTER, READ, or UPDATE—depends upon which facility you are using. (The ACF2 ADD is equivalent to the RACF ALTER.) Table 3–2 shows the type of access required to use Shadow facilities.

**Table 3–2. Shadow Access Requirements**

| Shadow Mainframe Adapter Server Facility                  | Suggested User        | Resources         | Access Required |
|---|-----------------------|-------------------|-----------------|
| Viewing product control blocks using ISPF/SDF Option 5.3. | DBA, Program Products | CONTROLBLOCK, SDB | READ            |
| Modifying product control blocks using a future facility. | DBA, Program Products | CONTROLBLOCK, SDB | UPDATE          |

**Table 3–2. Shadow Access Requirements (Continued)**

| <b>Shadow Mainframe Adapter Server Facility</b>   | <b>Suggested User</b>  | <b>Resources</b>            | <b>Access Required</b> |
|---|--|-----------------------------|------------------------|
| Using the SDB command.  | DBA, Program Products, VTAM, Operations                        | CONTROLBLOCK                | READ                   |
| Defining links using the <b>ADDRESS SDB DEFINE LINK</b> command.  | DBA, Program Products, VTAM, Operations                        | SDB                         | ADD/ALTER              |
| Viewing links using either ISPF/SDF Option 1 or the <b>ADDRESS SDB DISPLAY LINK</b> command.                              | DBA, Program Products, VTAM, Operations                        | LINKS                       | READ                   |
| Modifying links using either ISPF/SDF Option 1 or the <b>ADDRESS SDB MODIFY LINK</b> command.                             | DBA, Program Products, VTAM, Operations                        | LINKS, SDB                  | UPDATE                 |
| Defining databases using the <b>ADDRESS SDB DEFINE DATABASE</b> command.  | DBA, Program Products  | LINKS, SDB                  | ADD/ALTER              |
| Viewing databases using either ISPF/SDF Option 2 or the <b>ADDRESS SDB DISPLAY DATABASE</b> command.                      | DBA, Program Products, VTAM, Operations                        | DATABASES                   | READ                   |
| Modifying databases using either ISPF/SDF Option 2 or the <b>ADDRESS SDB MODIFY DATABASE</b> command.                     | DBA, Program Products  | DATABASES, SDB              | UPDATE                 |
| Viewing attached users using either ISPF/SDF Option 3 or the <b>ADDRESS SDB DISPLAY ATTACHED</b> command.                 | DBA, Program Products, VTAM, Operations                        | DATABASES, SDB              | READ                   |
| Viewing remote users using either ISPF/SDF Option 4 or the <b>ADDRESS SDB DISPLAY REMOTE</b> command.                     | DBA, Program Products, VTAM, Operations                        | USERS, SDB                  | READ                   |
| Killing remote users using ISPF/SDF Option 4.   | DBA, Operations, Developers, End-Users                         | USERS, SDB                  | READ, UPDATE           |
| Viewing product started task parameters using either ISPF/SDF Option 5.2 or the <b>ADDRESS SDB DISPLAY PARM</b> command.  | DBA, Program Products, VTAM, Operations                        | USERS, SDB                  | READ                   |
| Modifying product started task parameters using either ISPF/SDF Option 5.2 or the <b>ADDRESS SDB MODIFY PARM</b> command. | DBA, Program Products, VTAM, Operations                        | PARMS, SDB                  | UPDATE                 |
| Viewing all trace browse data.  | DBA, Program Products, VTAM, Operations                        | PARMS, SDB                  | READ                   |
| Issuing SQL statements via Shadow SPUFI.  | DBA, Program Products, VTAM, Operations                        | TRACEBROWSE, TRACEDATA, SDB | READ                   |
| Correlating userids or TCP/IP addresses to LTERMs.  | DBA, Shadow Administrator                                      | IMSLTERM, SDB               | READ, UPDATE           |
| Viewing global variables.   | All (DBA, Program Products, Operations, Developers, End-Users) | GLOBALS                     | READ                   |

**Table 3–2. Shadow Access Requirements (Continued)**

| Shadow Mainframe Adapter Server Facility | Suggested User                        | Resources    | Access Required                                  |
|--|---------------------------------------|--------------|--|
| Updating global variables.               | DBA, Shadow Administrator, Developers | GLOBALS      | UPDATE   |
| Refreshing Data Maps                     | DBA, Shadow Admin                     | SEF, DATAMAP | READ access to SEF;<br>UPDATE access to DATAMAP. |

## Defining Shadow Resources to RACF

Use the following steps to define classes and resources to RACF:

1. Define a new RACF class to the RACF Class Descriptor Table for RSDx, where x is the last character of the Shadow subsystem name.



**Note:**

Because RACF requires the class name to be a minimum of 4 characters, the class name must begin with the letter “R”. For additional information on how to add user-defined classes to the class descriptor table, please reference the *RACF System Programmer’s Guide*, “Chapter 3: RACF Customization.”

The following JCL can be used as a sample:

```
//STEP1      EXEC ASMHCL
//C.SYSLIB DD DSN=SYS1.MODGEN,DISP=SHR
//C.SYSIN DD *
RSDx      ICHERCDE CLASS=RSDx,
          ID=128,
          MAXLNTH=39,
          FIRST=ALPHANUM,
          OTHER=ANY,
          POSIT=25,
          OPER=NO
          ICHERCDE
/*
//L.SYSLMOD DD DSN=SYS1.LINKLIB,DISP=SHR
//L.SYSIN  DD *
          INCLUDE SYSLMOD( ICHRCDE )
          ORDER Rxyy
          ORDER *** Previous user-defined classes ***
          ORDER *** Previous user-defined classes ***
          ORDER ICHRCDE
          NAME ICHRCDE(R)
/*
```

2. Perform an IPL to change the RACF Class Descriptor Table. This is necessary for RACF to recognize the new class.

3. Activate the class to RACF with the following command:

```
SETROPTS CLASSACT(RSDx)
```

4. Define all RACF resource types to class RSDx with the following command:

```
RDEFINE RSDx CONTROLBLOCKS UACC(NONE)
```

Repeat this **RDEFINE** command for all RACF resource types.

5. Provide access to the resource according to the following example:

```
PERMIT CONTROLBLOCKS CLASS(RSDx) ID(AI38AAS) ACCESS(READ)
```

Where AI38AAS is the userid of the user to whom you wish to grant READ permissions.

Repeat this **PERMIT** command for all RACF resource types.

The NEON.CNTL(RACFDNF) member can be used as a sample for how to define the RACF class descriptor and router table. The NEON.CNTL(RACFSRC) member can be executed as a clist under TSO. It contains the **RDEFINE** and **PERMIT** statements, which will define the resource entities needed.

## Defining Shadow Resources to CA-Top Secret

Follow the steps below to define Shadow resources to CA-Top Secret:

1. Define an entry in the RDT, as shown in the following example:

```
TSS ADDTO(RDT) RESCLASS(SDx) RESCODE(nn) -  
  ATTR(LONG, PRIV, LIB, DEFPROT, GENERIC) -  
  ACLST(NONE, ALL, ALTER=1COO, UPDATE, READ) DEFACC(READ)
```

Where x is the last character of the Shadow subsystem name and nn is any hexadecimal code between 01 and 3F.



### **Note:**

When defining the CA-Top Secret class, you have to specify a parameter of LONG as shown in the above example.

2. Add all the resources to an owner with the following commands:

```
TSS ADDTO(owner) SDx(CONTROLBLOCKS)
```

Repeat this **TSS ADDTO** command for all resource types.

3. Permit the resources to profiles or users as follows:

```
TSS PERMIT(userid) SDx(TRACEDATA) ACC(READ)
```

## Defining Shadow Resources to ACF2

Use the following procedure to define Shadow resources to ACF2:

1. Define a generalized resource class named SDB.
2. Define resource rules for each of the resource classes that Shadow supports. Member ACF2DEFN of the NEON.CNTL dataset can be used as an example. The resource classes are as follows:

- CONTROLBLOCKS
- DATABASES
- LINKS
- PARMS
- SDB
- SEF
- TRACEBROWSE
- TRACEDATA
- USERS
- TOKENS
- RPC.<rpcname>

3. Use the following ACF2 command to allow users access to the resource rule:

```
ACFNRULE KEY(TRACEBROWSE) TYPE(SDx) ADD(UID(*****userid)
ALLOW
```

Where *x* is the last character of the Shadow subsystem name.

## Defining Shadow ISPF Load Modules

If you are using TSO Command Limiting to restrict execution access to TSO commands, you must define the Shadow ISPF load modules listed in Table 3–3 to your security product:

**Table 3–3. Shadow Load Modules**

|          |          |          |
|----------|----------|----------|
| SDADDM   | SDB2AUEx | SDRXIN   |
| SDADEX   | SDB2IN   | SDRXLELK |
| SDB      | SDB2RU   | SDRXPC   |
| SDBI     | SDDGRU   | SDRXSG   |
| SDBICOMP | SDDGSP   | SDRXSQ   |
| SDBIDB   | SDHOCM   | SDRXST   |
| SDBIMEX  | SDIMFU   | SDRXTK   |

**Table 3–3. Shadow Load Modules (Continued)**

|          |          |          |
|----------|----------|----------|
| SDBOB    | SDISCBRU | SDRXVA   |
| SDBOCP   | SDISSTRU | SDSLSVMD |
| SDBORU   | SDISTBRU | SDSLUTCC |
| SDBTIMD  | SDLEPLI  | SDSLUTCK |
| SDBVBFB  | SDLESVRU | SDSLUTDE |
| SDBX     | SDLINK   | SDSLUTKY |
| SDBXCOMP | SDNTLMD  | SDSLUTPA |
| SDBXDB   | SDRXBR   | SDSLUTRQ |
| SDBXSCAN | SDRXID   |          |

## Using the RACF Pass Ticket

The RACF Pass Ticket can be used instead of a user's logon password. When you use a RACF Pass Ticket with Shadow, the application name passed is the 3-character subsystem ID code (e.g, SDB for Shadow Mainframe Adapter Server and SWS for Shadow Web Server) appended with the system SMFID. This application name must match a PTKTDATA profile name for Pass Ticket generation and authentication to work. For example, if the system SMFID is DEV1, the application name will be SDBDEV1, and you will need to define a PTKTDATA profile for Shadow with the name SDBDEV1.

Also, a PTKTDATA profile name can be further qualified by RACF userid and/or RACF connect group (for example, SDBDEV1.SDBB or SDBDEV1.SYS1.SDBB). This allows different instances of an application to have their own unique SSO keys.

For more information on defining profiles in the PTKTDATA class, please see the IBM RACF Manual. However, for your convenience, the following the syntax from the IBM RACF Manual is provided below for defining a profile name in the PTKTDATA class:

### *5.13.3 Defining Profiles in the PTKTDATA Class*

*For each application that users can gain access to with the PassTicket, you must create at least one profile in the PTKTDATA class. The profile associates a secret Secured Signon application key with a particular application on a particular system. The profiles can be created so they apply to:*

- *All users who need access to the application*
- *A specific RACF group of users who need access to the application*
- *A specific RACF user, when connected to a specific RACF group*
- *A specific RACF user*

*To define the profile, use the RDEFINE command:*

```
RDEFINE PTKTDATA profile_name
      SSIGNON(key_description)
      UACC(access_authority)
```

where:

**PTKTDATA**

*specifies the PassTicket Key class.*

**profile\_name**

*is the name of the profile (see "Determining Profile Names" in topic 5.13.3.1).*

*For the PTKTDATA class, the profile must be a discrete profile. Because each application must be uniquely defined, you cannot specify a generic profile in the PTKTDATA class. If you specify a generic profile, it is ignored during PassTicket processing for the application, and PassTickets cannot be used to authenticate users for that application.*

**key\_description**

*defines the Secured Signon application key and specifies the method RACF is to use to protect it in the RACF database on the host. You can specify either masking or encryption for the method (see "Protecting the Secured Signon Application Keys" in topic 5.13.3.2).*

*Secured Signon keys are 64-bit Data Encryption Standard (DES) keys. With DES, 8 of the 64 bits are reserved for use as parity bits, so those 8 bits are not part of the 56-bit key. In hexadecimal notation, the DES parity bits are: X'0101 0101 0101 0101'.*

*Any two 64-bit keys are equivalent DES keys if their only difference is in one or more of these parity bits.*

**access\_authority**

*is the universal access authority to be associated with the resource protected by this profile. By default, the UACC is NONE for the PTKTDATA class.*

*After a profile in the PTKTDATA class has been created, you can change it with the RALTER command, which is similar in syntax to the RDEFINE command:*

```
RALTER PTKTDATA profile_name
      SSIGNON(key_description)
      UACC(access_authority)
```

## Note on Started Task Security

A major exception to the security authorization scheme is the Shadow Mainframe Adapter Server started task itself. All work performed under the product address space, on behalf of the started task, is exempt from security. As a practical matter, this means that the SDBB address space itself does not need authorization to run its own initialization EXEC or manipulate the SEF rulesets. All work performed

within the product address space on behalf of external client requests is subject to security authorization checking.

## Controlling Information Access with the TRACEDATA Resource

The TRACEDATA resource controls access to two types of information contained within the Shadow Mainframe Adapter Server trace log:

- SQL source statements (the real SQL source statements, as taken from DBRMs or prepared strings, which may contain table names, column names, etc.)
- Binary data that underlies the trace log

Users who have READ authority for the TRACEDATA resource (as well as READ authority for SDB and TRACEBROWSE) are permitted to view the trace log information in its entirety. Users who don't have READ authority have only restricted access to this information.

The TRACEDATA resource restricts data differently, depending on the type of event:

- **SQL Events:** If your userid matches the userid associated with the event, you are permitted to look at an uncensored log of the SQL event. Otherwise, you can only see a censored representation of the SQL statement. The censored version includes the SQL verb but does not include table names, column names, etc.
- **Non-SQL Events:** If your userid matches the userid associated with the event, you are permitted to see an uncensored view of the underlying binary data for event. Otherwise, you are not allowed to see the binary data at all; no data is displayed and a message is written to the terminal.

## Resource Security for Test Versions of Shadow Mainframe Adapter Server

All resource security is simulated for test versions of Shadow Mainframe Adapter Server running in a TSO session. The z/OS security subsystem is not actually consulted, since a test TSO copy of the product is not authorized to perform this type of security check, and all work is performed using the TSO user's existing z/OS authorizations.

In this environment, all security checks are assumed to have completed successfully. If you are running test copies of the Shadow under TSO, you should find this feature helpful in deploying new applications, since you can review the security checks that will occur when the application is deployed in a production environment.



# CHAPTER 4:

# **Shadow Mainframe Adapter Server: Connecting to TSO**

---

---

This chapter covers the steps for setting up Shadow Mainframe Adapter Server to run under TSO. Shadow Mainframe Adapter Server is the server component of the Shadow product.

Topics covered include the following:


- Setting Up Shadow Mainframe Adapter Server to Run Under TSO
- Running a Test Version

## **Setting Up Shadow Mainframe Adapter Server to Run Under TSO**

Before running a Shadow Mainframe Adapter Server session under a TSO user's address space, the TSO user must be set up to run exactly as the server:

1. Allocate all of the Shadow ISPF datasets to the user's logon proc as follows:

|          |   |
|----------|---|
| ISPLLIB  | NEON.SV040800.LOAD  |
| ISPMLIB  | NEON.SV040800.NEONMLIB                                    |
| ISPPPLIB | NEON.SV040800.NEONPLIB                                    |
| ISPTLIB  | NEON.SV040800.NEONTLIB                                    |
| SYSEXEC  | NEON.SV040800.EXEC(FB) ( <i>FB if using FB datasets</i> ) |
| SDBTRACE | ( <i>Optional dd statement, see #2 below</i> )            |
| SDBRPCLB | ( <i>Optional dd statement, see #6 below</i> )            |

 **Note:**

If you are using a Shadow Load Library that is **not** APF authorized, you will need to copy NEON.SV040800.LOAD to that version of the load library and use that dataset instead on the ISPLLIB allocations.

2. **(Optional)** Allocate a new trace file as follows:
  - a. Use job NEON.SV040800.CNTL(DEFDIV) to allocate the linear trace dataset.
  - b. Allocate this linear trace dataset to the user's ddname (SDBTRACE). It is recommended that you only allocate a small trace dataset.



**Note:**

If you do not create a trace file, all trace information is lost during shutdown.

3. **(Optional)** In a library allocated to SYSEXEC, do one of the following:
  - Customize a copy of the existing initialization EXEC (see “Step 9: Customize the Initialization EXEC.” on page 2-14 within Chapter 2, “Shadow Server: Planning and Installation,” of this guide).
  - Create a new initialization EXEC (SDBxIN00, where x is the 4th character of the new subsystem). This initialization EXEC should be set up with only the minimal parameters. The fewer the parameters, the quicker the test server will initialize.



**Note:**

If you create a new initialization EXEC, Shadow recommends that the BROWSEMAX parameter be set at 10,000.

4. If you are using TCP/IP, define a new port number in the initialization EXEC for TCP/IP connections.
5. If you are using LU6.2, define and use a new APPLID.
6. To run RPCs, allocate your RPC load library to a ddname of SDBRPCLB.

You are now ready to verify the installation by running a test version.

## Running a Test Version

After setting up the user’s TSO address space, you will need to start a test version to verify the installation as follows:

1. Log on to a TSO/ISPF session. The system displays the **ISPF Primary Option Menu** panel, shown in Figure 4–1.

```

Menu  Utilities  Compilers  Options  Status  Help

                                ISPF Primary Option Menu

Option ==>

0  Settings      Terminal and user parameters      User ID . :
1  View          Display source data or listings   Time. . . :
2  Edit          Create or change source data      Terminal. :
3  Utilities     Perform utility functions        Screen. . :
4  Foreground   Interactive language processing   Language. :
5  Batch         Submit job for language processing Appl ID . :
6  Command      Enter TSO or Workstation commands TSO logon :
7  Dialog Test  Perform dialog testing            TSO prefix:
8  LM Facility  Library administrator functions  System ID :
9  IBM Products IBM program development products MVS acct. :
10 SCLM         SW Configuration Library Manager  Release . :
11 Workplace   ISPF Object/Action Workplace
I  Installation Installation Applications
M  More         Additional IBM Products

Enter X to Terminate using log/list defaults

```

**Figure 4–1. ISPF Primary Option Menu**

2. From this menu, select Option 6, Command.
3. Press ENTER. The system displays the **ISPF Command Shell** panel, shown in Figure 4–2.

```

Menu  List  Mode  Functions  Utilities  Help

                                ISPF Command Shell

Enter TSO or Workstation commands below:

====> _____
_____
_____

Place cursor on choice and press enter to Retrieve command

```

**Figure 4–2. ISPF Command Shell**

4. Type the **SDB** command followed by the name of the subsystem created in the new initialization EXEC. This is the one you created when you set up the server to run under TSO (see “Setting Up Shadow Server to Run Under TSO” on page 4-1). The command should be typed as follows:

```
SDB SUB (SDBx)
```

Where x is the 4th character of the subsystem name.

5. Press ENTER. The system displays the **Shadow Mainframe Adapter Server Primary Option Menu** panel, shown in Figure 4–3.

```

----- Shadow Mainframe Adapter Server Primary Option Menu -----
-----
Option ==>

  1 LINK          - Display and control link table           Time      - 10:22
  2 IMS           - IMS Control Facility                     Terminal - 3278
  3 CICS          - CICS Control Facility                   PF Keys  - 24
  4 REMOTE USER  - Display and control remote users         VV.RR.MM - 04.05.01
  5 SDB CONTROL  - Control Shadow Mainframe Adapter Server

Subsys  - SDBB
  6 TRACE BROWSE - Browse Shadow Mainframe Adapter Server trace log
  7 SEF CONTROL  - Control Shadow Event Facility (SEF)
  8 DATABASES    - Monitor and control database access
 10 DATA MAPPING - Data Mapping Facility
 11 ACI          - Advanced Communications Interface
  D DEBUG        - Debugging Facilities
  S SUPPORT      - Display Shadow Mainframe Adapter Server Support Informa

```

**Figure 4–3. Shadow Mainframe Adapter Server Primary Option Menu**

6. From this menu, select Option D, Debug.
7. Press ENTER. The system displays the **Shadow Mainframe Adapter Server Debugging Menu** panel, shown in Figure 4–4.

```

----- Shadow Mainframe Adapter Server Debugging Menu -----
-----
OPTION ==>

  1 SDB Test      - Start or Stop a test copy of the product
  2 SDB Debug     - Debug Stored Procedures

These functions operates only for test copies of the product started
under TSO/E. Consult the installation reference manual for instructions

```

**Figure 4–4. Shadow Mainframe Adapter Server Debugging Menu**

8. From the menu, select Option 1, SDB Test.
9. Press ENTER. The system displays the **Shadow Mainframe Adapter Server Debugging Control** panel, as shown in Figure 4–5.

```

----- Shadow Mainframe Adapter Server Debugging Control -----
-----
OPTION  ==>

      S - Start test copy of SDB                P - Stop test copy of SDB

OPDBIN START-UP PARAMETERS:

OPTION          ==> INIT      (INIT,DINT,DCES,DQPC,DQPT,DISN,DISC)
SUBSYSTEM NAME ==> SDBB      (SDBx, "x" must be between A - Z)

```

**Figure 4–5. Shadow Mainframe Adapter Server Debugging Control**

10. On the option line, type S to start the server.
11. Under **OPDBIN START-UP PARAMETERS**, in the **SUBSYSTEM NAME** field, type the subsystem name for the subsystem that you created when you set up the server to run under TSO.
12. Press ENTER. You should get the following message if the Shadow Mainframe Adapter Server was installed correctly:

```
SUBSYS SDBX INITIALIZATION COMPLETE
```

13. Press F3 to return to the **Shadow Mainframe Adapter Server Debugging Control** panel (Figure 4–5).
14. On the option line, type P to stop the Shadow Mainframe Adapter Server.
15. Press ENTER. You should get the following message if the Shadow Mainframe Adapter Server was terminated successfully:

```
SUBSYS SDBX TERMINATION COMPLETE
```



# Shadow Interface for Natural Installation

---





# CHAPTER 5: *Shadow Interface for Natural: Planning and Installation*

---

---

This chapter describes planning and installation required for the Shadow Interface™ for Natural, part of the Shadow product.

Topics include the following:

- Installation Prerequisites
- Installation Steps
  - Installation for CICS
  - Installation for Batch

## **Installation Prerequisites**

Before using the Shadow Interface for Natural, you must meet the following:

- The Shadow Mainframe Adapter Server component has been installed.
- The Shadow Interface for CICS/TS has been installed.
- The Shadow Mainframe Adapter Client component has been installed and configured.
- The Shadow product has been licensed for the following components:
  - Shadow Interface for Natural.
  - Shadow Interface for CICS/TS
- The minimum version of Natural is Natural 2.2.8.

## **Installation Steps**

The installation depends on whether you are using the Shadow Interface for Natural in a CICS or a batch environment.

### ***Installation for CICS***

The following steps are required to install the Shadow Interface for Natural for CICS:

- Step 1: Activate the SDBMAPP DD Statement
- Step 2: Assemble the ACI Translation/Migration Table
- Step 3: **(Optional)** Change the Default Name in the SDCICIS Stub
- Step 4: Link the Natural Nucleus with the SDCICIS Stub
- Step 5: Create the CICS Resource Definitions

- Step 6: Assemble the CICS PLTPI for ACI Task Related User Exit
- Step 7: Make the CICSLOAD Dataset Modules Available to CICS
- Step 8: Restart the CICS Region
- Step 9: Verify the Installation

## Step 1: Activate the SDBMAPP DD Statement

Ensure that the SDBMAPP DD statement is defined in the Shadow started task JCL.



### **Doc Reference:**

For more information about defining the SDBMAPP DD statement in the Shadow started task JCL, see “Step 6: Set Up the Started Task JCL.” on page 2-8 of this guide.

## Step 2: Assemble the ACI Translation/Migration Table

The Shadow Interface for Natural ACI translation/migration table (SDBRTX table) allows you to do the following:

- **(Required)** Assign a default Shadow Mainframe Adapter Server subsystem to route all service application ACI requests.



### **Note:**

The default Shadow Mainframe Adapter Server subsystem can be overridden by using the ACI-ID field in the REGISTER function to specify a Shadow Mainframe Adapter Server subsystem.

- Route the specified CICS transaction IDs to specific Shadow Mainframe Adapter Server subsystems.
- (If you are a current Software AG Broker ACI user) Migrate to the Shadow Interface for Natural.
- Specify Natural parameters.

### **Use of the SDBRTX Table**



### **Note:**

This module is *not* a CICS program but is loaded into the CICS region as such.

This module is used as follows:

- It is initially loaded by SDCITRUI (the ACI Task Related User Exit Initialization program).

- It is viewed by SDCITRU (the ACI Task Related User Exit).
- If SDCIFEN is not used to pass data to the transaction (see “Defining Startup Parameters” on page 5-3) and the CICS transaction is associated with the SDCINAT Natural “front end” program, then SDCINAT will use the ACI translation/migration table (the SDBRTX table) to obtain information to start the Natural session.

## Defining Startup Parameters

Transaction startup parameters can be defined as follows:

- **Using the SDBRTX table.** Transaction startup parameters can be defined in the SDBRTX entry.
- **Using the SDCIFEN program.** SDCIFEN is a CICS module that allows for the passing of the COMMAREA to the invoked ACI server if the CICS transaction is associated with the SDCIFEN program. If you associate the transaction name with the program SDCIFEN when defining the transaction to CICS, then you may pass Natural startup parameters to the transaction using the COMMAREA. In turn, SDCIFEN transfers control (EXEC CICS XCTL) to a program of your choice.

If SDCINAT is used as the program to which SDCIFEN transfers control, the COMMAREA must be formatted with the Natural nucleus program name (8 bytes, space padded if necessary), followed by a comma, followed by the CICS transaction ID for Natural (4 bytes, space padded if necessary), followed by a comma, followed by the Natural startup parameters.



### Note:

Even if the SDCIFEN program will be used to specify the Natural startup parameters, as a minimum, the SDBRTX table *must* be assembled with the TYPE=INITIAL card with a default Shadow subsystem defined and a TYPE=FINAL card to set the default subsystem.



### Doc Reference:

For more information about using SDCIFEN to define Natural startup parameters, see step 7 of “Shadow Interface for Natural: Administration,” on page 4-5.

## Assembling the SDBRTX Table

The Shadow Mainframe Adapter Server SAMP dataset member ACI1 contains the source that needs to be assembled for the ACI translation/migration table. The ACI translation/migration table should be installed into a load library that is concatenated to the DFHRPL DD statement. This can include the CICSLOAD dataset that has been downloaded from the Shadow installation tape.

The minimal assembled module **must** contain the TYPE=INITIAL card with a default Shadow subsystem defined and a TYPE=FINAL card to set the default subsystem.

**Note:**

If you want to run multiple concurrent Natural sessions, you must set the ETID and AUTO parameters to OFF.

**Example**

The following example shows an example SDBRTX entry, set up for running multiple sessions under the same CICS connection:

```

REQUIRED--> ACI    TYPE=INITIAL, SUBS=SDBB           DEFINITION START / DEFAULT
ACI    TYPE=TRANS, TRAN=ACI1, SUBS=SDBB TRANSACTION DEFINITION
ACI    TYPE=ACI, ETBBRID= 'XXXXXXXXXXXXXXXXXXXXXXX',           X
        SUBS=SDBB           ACI TRANSLATION
ACI    TYPE=NATURAL, TRAN=ACI1, SUBS=SDBB, NATPGM=NC313RE,       X
        NATPRM= 'MENU=OFF, SENDER=NLST, OUTDEST=NLST, ETID=OFF,   X
        AUTO=OFF, STACK=(LOGON ACI,userid,password;IVP;FIN) ',   X
        NATTRN=NATL
ACI    TYPE=FINAL           DEFINITION END

```

**Note:**

NC313RE is the nucleus that is linked in “Step 4: Link the Natural Nucleus with the SDCICIS Stub” on page 5-5.

### Step 3: (Optional) Change the Default Name in the SDCICIS Stub

**Note:**

This step is **only** required for users who wish to run the Shadow Interface for Natural **concurrently** with SAG EntireX.

By default, the Shadow Interface for Natural allows users to run Natural programs that were created to run with SAG EntireX unchanged in Shadow. To do this, the Shadow Interface for Natural intercepts the CALL 'BROKER' calls to EntireX and routes them to Shadow instead.

If you wish to run the Shadow Interface for Natural concurrently with SAG EntireX, you must re-linkedit the SDCICIS stub to change the default name from BROKER to SHADOW. When you do this, the Shadow Interface for Natural will process CALL 'SHADOW' calls instead of CALL 'BROKER' calls, so Natural programs that use EntireX will continue to use EntireX.

**Note:**

After changing the default name from BROKER to SHADOW, if you want your Natural programs to use the Shadow Interface for Natural instead of EntireX, you must edit your Natural programs and change the CALL 'BROKER' calls to CALL 'SHADOW' calls.

**Example**

The following example JCL shows how to re-linkedit the SDCICIS stub to change the default name from BROKER to SHADOW.

```
...jobcard....
//L          EXEC PGM=HEWL,PARM='MAP,LET,LIST'
//SYSLIN    DD  DDNAME=SYSIN
//NEONLIB   DD  DSN=your-shadow-cicsload,DISP=SHR
//SYSLMOD   DD  DSN=your-shadow-cicsload,DISP=SHR
//SYSUT1    DD  DSN=&&SYSUT1,SPACE=(1024,(120,120),,,ROUND),
//          UNIT=VIO,DCB=BUFNO=1
//SYSPRINT  DD  SYSOUT=*
//SYSIN     DD  *
              CHANGE BROKER(SHADOW)  !--CHANGING THE DEFAULT NAME
              INCLUDE NEONLIB(SDCICIS)
              NAME SDCICIS(R)
//
```

**Step 4: Link the Natural Nucleus with the SDCICIS Stub**

Linkedit the SDCICIS stub into your Natural CICS nucleus.

**Example**

```
//NEONLIB DD DSN=NEON.CICSLOAD,DISP=SHR
.
.
INCLUDE NEONLIB(SDCICIS)  FOR SHADOW ACI STUB
INCLUDE NATLIB(NATLAST)  FINAL INCLUDE
```

**Note:**

If you changed the default name from BROKER to SHADOW in the SDBCICIS stub (see “Step 3: (Optional) Change the Default Name in the SDCICIS Stub” on page 5-4), you must add SHADOW as a CSTATIC parameter. An example is as follows:

```
CSTATIC=(CMMSG,SHADOW)
```

## Step 5: Create the CICS Resource Definitions

The Shadow Interface for Natural requires several resources that include programs, a transaction, and an intrapartition transient data queue (TDQ) to be defined in all CICS regions that will be using this component.

The Shadow Mainframe Adapter Server CNTL dataset member CICSACI is a sample JCL you can use to help you create these definitions. Refer to this member for further instructions.

## Step 6: Assemble the CICS PLTPI for ACI Task Related User Exit

1. After defining SDCITRUI and SDCITRU to CICS (in “Shadow Interface for Natural: Planning and Installation,”), add the following entry to the program list table/program initialized (PLTPI) table:

```
DFHPLT TYPE=ENTRY, PROGRAM=SDCITRUI
```



**Note:**

This entry should *follow* the first of the following statements to ensure that the SDCITRUI will be executed in phase 2 of the CICS PLTI process:

```
DFHPLT TYPE=ENTRY, PROGRAM=DFHDELIM
```

2. You must code the appropriate PLTPI=xx parameter in CICS startup, where xx is the suffix value given to the DFHPLTI table.

## Step 7: Make the CICSLOAD Dataset Modules Available to CICS

You can perform this step in one of the two following ways:

- (Recommended method) Add the CICSLOAD dataset to the CICS DFHRPL dataset concatenation.
- Copy the CICSLOAD dataset members to another dataset that is already in the CICS DFHRPL dataset concatenation.



**Note:**

The CICSLOAD dataset should have been unloaded during the regular Shadow Mainframe Adapter Server installation process, after you submit the INSTALL job to unload Shadow Mainframe Adapter Server’s distribution files. The default name of this dataset is NEON.CICSLOAD. For more information, see Chapter 2, “Shadow Server: Planning and Installation,” of this guide.

The first method is the recommended one for ease of administration.

## Step 8: Restart the CICS Region

Since the CICS installation for the Shadow Interface for Natural requires use of a CICS Task Related User Exit, you should restart your CICS region before attempting to use it.

The following message should appear on the console if the CICS Task Related User Exit correctly installs and is enabled:

```
SDB0364I SDCITRU is started and enabled.
```

## Step 9: Verify the Installation

To perform the Shadow Interface for Natural installation verification process (IVP) for CICS, complete the following steps to create a service definition:

1. Upload the sample IVP program.

Load the Shadow Interface for Natural sample IVP program for CICS using member ACI2 in the Shadow Mainframe Adapter Server SAMP dataset. Make sure you have coded a TYPE=NATURAL entry during the installation procedure, as described in “Step 2: Assemble the ACI Translation/Migration Table” on page 5-2.

2. From the **Shadow Mainframe Adapter Server Primary Option Menu**, select Option 11.1.1.
3. Press ENTER. The system will display the **Shadow Mainframe Adapter Server ACI Extract** panel, shown in Figure 5–1.

```
----- Shadow Mainframe Adapter Server ACI Extract -----
-----
COMMAND ===>

Map Dataset Library:
  Project . . . _____
  Group . . . . _____
  Type . . . . _____
  Member . . . . _____

Other Map Dataset Name:
```

**Figure 5–1. Shadow Mainframe Adapter Server ACI Extract Panel**

4. Specify the information required for the map dataset library. You can type the **Project**, **Group**, **Type**, and **Member** names in the fields provided; alternatively, you can specify the dataset name in the **Other Map Dataset Name** field.

**Notes:**

- The dataset must have already been assigned to SDBMAPP.
- The Shadow Interface for Natural sample IVP program for CICS will be named ACICNIVP.

5. Press ENTER. The system displays the **Shadow Mainframe Adapter Server CICS ACI Extract** panel, as shown in Figure 5–2.

```

----- Shadow Mainframe Adapter Server CICS ACI Extract -----
-----
COMMAND ===>

Server Name . . . . . ACICNIVP (R)
Server Service Class. _____ (O)
Server Service. . . . _____ (O)

Persistant Connection N (Y/N) (R) Unit of Work Participant N (Y/N) (R)
Secure this Service . N (Y/N) (R) Maximum UOW Buffer Size 0 (O)
Mirror Transaction. . EXCI (R) Max Execution Time (secs) 0 (O)
Connection name . . . EXCS (R)
Transaction Name. . . ACIF (R)
Max Allowed . . . . . 10 (1-9999 servers) (R)
Auto Terminate. . . . 100 (0-99999 Receives, Persist N ONLY) (O)
Client Non-Activity Timer . . . . 00 : 01 : 00 (hh:mm:ss) (O)
Server Shutdown Non-Activity Timer 00 : 01 : 00 (hh:mm:ss) (O)
Maximum Wait for Server Timer . . 00 : 00 : 00 (hh:mm:ss) (O)

If using SDCIFEN as the program associated with the Transaction Name
above, SDCIFEN will XCTL control to program SDCINAT
with a commarea containing: NC313RE ,NATL,MENU=OFF,SENDER=NLST,OUTDEST=NLST,
ETID=OFF,AUTO=OFF,MADIO=0,STACK=(LOGON ACI,userid,password;ACICNIVP;FIN)

```

**Figure 5–2. Shadow Mainframe Adapter Server CICS ACI Extract**

6. Specify the following information:

- **Server Name**
- **Server Service Class**
- **Server Service**

**Note:**

The **Server Name** must be coded exactly as shown in Figure 5–2 in order for the IVP program to work. The “triple name” (which is the combination of the **Server Name**, **Server Service Class**, and the **Server Service**) are imperative for the association of the IVP program to Shadow during server registry.



- **Mirror Transaction**
- **Connection Name**
- **Transaction Name**

**Note:**

The **Mirror Transaction**, **Connection Name**, and **Transaction Name** are defined by the customer for the transaction to be invoked under CICS and have previously been defined in the Shadow EXCI.

7. Press ENTER to perform the extract. If the extract was successful, the system will display the message “Service is now defined.”
8. Use the **END** command (or press F3) to return to the **Shadow Mainframe Adapter Server ACI Facility** options menu (Figure 5–3).

```

----- Shadow Mainframe Adapter Server ACI Facility -----
---- Subsystem SDBB
OPTION ===>

 1 ACI Server Definition - Create ACI Server Map Information
 2 Natural Extract       - Create Natural Map Information
 3 ACI Map Display      - Display ACI Server Map Information
 4 Active Server Display - Display Active ACI Servers
 5 Map Refresh          - Refresh Shadow Maps
 6 ACI Error Create     - Create ACI Error Processing Definitions
 7 ACI Error Display    - Display ACI Error Processing Definitions

```

**Figure 5–3. Shadow Mainframe Adapter Server ACI Facility Options Menu**

9. From this menu, select Option 5, Map Refresh.
10. Press ENTER to refresh the service definition into Shadow.
11. From this menu, select Option 3, ACI Map Display.
12. Press ENTER to display the member name (in this example, ACICNIVP) as a structure name.
13. Within your Java program, issue the following SQL statement:

```
CALL SHADOW_ACI( 'SEND' , 'ACICNIVP' , 'DATA' )
```

*Doc Reference:*

For more information about performing a query, see the *Shadow Mainframe Adapter Client for Natural Client Installation and Administration Guide*.

## Installation for Batch

The following steps are required to install the Shadow Interface for Natural for batch:

- Step 1: Activate the SDBMAPP DD Statement
- Step 2: Assemble the ACI Translation/Migration Table
- Step 3: **(Optional)** Change the Default Name in the SDBRBIS Stub
- Step 4: Link the Natural Batch Nucleus with the Batch SHADOW\_ACI Stub
- Step 5: Modify the Shadow Mainframe Adapter Server Started Task JCL
- Step 6: Verify the Installation

### Step 1: Activate the SDBMAPP DD Statement

Ensure that the SDBMAPP DD statement is defined in the Shadow started task JCL.



#### **Doc Reference:**

For more information about defining the SDBMAPP DD statement in the Shadow started task JCL, see “Step 6: Set Up the Started Task JCL.” on page 2-8 of this guide.

### Step 2: Assemble the ACI Translation/Migration Table into the STEPLIB Dataset

The Shadow Interface for Natural ACI translation/migration table (SDBRTX table) allows you to do the following:

- **(Required)** Assign a default Shadow Mainframe Adapter Server subsystem to route all service application ACI requests.



#### **Note:**

The default Shadow Mainframe Adapter Server subsystem can be overridden by using the ACI-ID field in the REGISTER function to specify a Shadow Mainframe Adapter Server subsystem.

- (If you are a current Software AG Broker ACI user) Migrate to the Shadow Interface for Natural.

### **Assembling the SDBRTX Table in the STEPLIB Dataset**

The Shadow Mainframe Adapter Server SAMP dataset member ACI1BAT contains the source that needs to be assembled for the ACI translation/migration table.

You must place the SDBRTX module in the STEPLIB of the started task procedure that you use.

▷ **Note:**  
For each Shadow Mainframe Adapter Server subsystem, you need to create a separate SDBTRX module located in separate load library.

The minimal assembled module *must* contain the TYPE=INITIAL card with a default Shadow subsystem defined and a TYPE=FINAL card to set the default subsystem.

### Example

The following example shows an example SDBRTX entry:

```
REQUIRED--> ACI    TYPE=INITIAL, SUBS=SDBB  DEFINITION START / DEFAULT
ACI    TYPE=ACI, ETBBRID= 'XXXXXXXXXXXXXXXXXXXXXXX',           X
          SUBS=SDBB                      ACI TRANSLATION
ACI    TYPE=FINAL                      DEFINITION END
```

## Step 3: (Optional) Change the Default Name in the SDBRBIS Stub

▷ **Note:**  
This step is *only* required for users who wish to run the Shadow Interface for Natural *concurrently* with SAG EntireX.

By default, the Shadow Interface for Natural allows users to run Natural programs that were created to run with SAG EntireX unchanged in Shadow. To do this, the Shadow Interface for Natural intercepts the CALL 'BROKER' calls to EntireX and routes them to Shadow instead.

If you wish to run the Shadow Interface for Natural concurrently with SAG EntireX, you must re-linkedit the SDBRBIS stub to change the default name from BROKER to SHADOW. When you do this, the Shadow Interface for Natural will process CALL 'SHADOW' calls instead of CALL 'BROKER' calls, so Natural programs that use EntireX will continue to use EntireX.

▷ **Note:**  
After changing the default name from BROKER to SHADOW, if you want your Natural programs to use the Shadow Interface for Natural instead of EntireX, you must edit your Natural programs and change the CALL 'BROKER' calls to CALL 'SHADOW' calls.

## Example

The following example JCL shows how to re-linkedit the SDBRBIS stub to change the default name from BROKER to SHADOW.

```
...jobcard....
//L          EXEC PGM=HEWL,PARM='MAP,LET,LIST'
//SYSLIN    DD  DDNAME=SYSIN
//NEONLIB   DD  DSN=your-shadow-neon.load,DISP=SHR
//SYSLMOD   DD  DSN=your-shadow-neon.load,DISP=SHR
//SYSUT1    DD  DSN=&&SYSUT1,SPACE=(1024,(120,120),,ROUND),
//          UNIT=VIO,DCB=BUFNO=1
//SYSPRINT  DD  SYSOUT=*
//SYSIN     DD  *
            CHANGE BROKER(SHADOW)  <--CHANGING THE DEFAULT NAME
            INCLUDE NEONLIB(SDBRBIS)
            NAME SDBRBIS(R)
//
```

## Step 4: Link the Natural Batch Nucleus with the Batch SHADOW\_ACI Stub

Link the batch Natural nucleus with the SHADOW\_ACI stub module, SDBRBIS.

### Example

```
//NEONLIB DD DSN=NEON.LOAD
.
.
INCLUDE NEONLIB(SDBRBIS)
INCLUDE NATLIB
```



### Note:

If you changed the default name from BROKER to SHADOW in the SDBRBIS stub (see “Step 3: (Optional) Change the Default Name in the SDBRBIS Stub” on page 5-11), you must add SHADOW as a CSTATIC parameter. An example is as follows:

```
CSTATIC=(CMMSG,SHADOW)
```

## Step 5: Modify the Shadow Mainframe Adapter Server Started Task JCL

Ensure that the Shadow Mainframe Adapter Server startup JCL contains the following DD statement:

```
/******//
//* THIS DD IS FOR WRITING SERVER JCL TO THE INTERNAL READER*//
//*****//
//SDBINRDR DD SYSOUT=(,INTRDR),
// DCB=(RECFM=F,LRECL=80,BLKSIZE=80,DSORG=PS)
```

## Step 6: Verify the Installation

To perform the ACI installation verification process (IVP) for ACI batch, complete the following steps:

1. Upload the sample IVP program.

Load the Natural ACI sample IVP program using member ACI2BAT in the Shadow Mainframe Adapter Server SAMP dataset.

2. Modify the batch JCL, as follows:

- **Submitting Service as Batch Job:** Member ACIJCL1 in the Shadow Mainframe Adapter Server SAMP library contains sample JCL to execute sample program as a batch job. Modify this JCL to your company standards and requirements.
- **Submitting Service as Started Task:** Member ACIJCL2 in the Shadow Mainframe Adapter Server SAMP library contains sample JCL to execute sample program as a started task. Modify this JCL to your company standards and requirements and copy it into the startup PROCLIB.

3. From the **Shadow Mainframe Adapter Server Primary Option Menu**, select Option 11.1.2.

4. Press ENTER. The system will display the **Shadow Mainframe Adapter Server ACI Extract** panel, shown in Figure 5–1.

```

----- Shadow Mainframe Adapter Server ACI Extract -----
-----
COMMAND ===>

Map Dataset Library:
  Project . . . _____
  Group . . . . _____
  Type . . . . _____
  Member . . . . _____

Other Map Dataset Name:

```

**Figure 5–4. Shadow Mainframe Adapter Server ACI Extract Panel**

5. Specify the information required for the map dataset library. You can type the **Project**, **Group**, **Type**, and **Member** names in the fields provided; alternatively, you can specify a data set name in the **Other Map Dataset Name** field.

**Notes:**

- The dataset represented by the project name, group name, and type must have already been assigned to SDBMAPP.
- The Shadow Interface for Natural sample IVP program for batch will be named ACIBNIVP.

6. Press ENTER. The system displays the **Shadow Mainframe Adapter Server Batch ACI Extract** panel, as shown in Figure 5–2 and Figure 5–6.

```

----- Shadow Mainframe Adapter Server Batch ACI Extract -----
-----
COMMAND ==>>

Server Name . . . . . ACIBNIVP (R)
Server Service Class. _____ (O)
Server Service. . . . _____ (O)

JCL DSN (no quotes) . YOUR.NEON.CNTL(ACIBNIVP) (R)
--- OR ---
Console Command . . . _____ (R)

Max Allowed . . . . . 1_____ (1-9999 servers) (R)
Persistant Connection N (Y/N) (R) Unit of Work Participant N (Y/N)
Secure this Service . N (Y/N) (R) Maximum UOW Buffer Size _____
Auto Terminate. . . . 0_____ (0-99999 Receives, Persist N ONLY) (O)
Client Non-Activity Timer . . . . 00:00:30 (hh:mm:ss) (O)

```

**Figure 5–5. Shadow Mainframe Adapter Server Batch ACI Extract -- Submitting Service as Batch**

```

----- Shadow Mainframe Adapter Server Batch ACI Extract -----
-----
COMMAND ==>>

Server Name . . . . . ACIBNIVP (R)
Server Service Class. _____ (O)
Server Service. . . . _____ (O)

JCL DSN (no quotes) . _____ (R)
--- OR ---
Console Command . . . S SDBBNAT,PROG=ACIBNIVP (R)

Max Allowed . . . . . 1_____ (1-9999 servers) (R)
Persistant Connection N (Y/N) (R) Unit of Work Participant N (Y/N)
Secure this Service . N (Y/N) (R) Maximum UOW Buffer Size _____
Auto Terminate. . . . 0_____ (0-99999 Receives, Persist N ONLY) (O)
Client Non-Activity Timer . . . . 00:00:30 (hh:mm:ss) (O)

```

**Figure 5–6. Shadow Mainframe Adapter Server Batch ACI Extract -- Submitting Service as Started Task**

7. Specify the following information:

- **Server Name**
- **Server Service Class**
- **Server Service**



**Note:**

The **Server Name** must be coded exactly as shown in Figure 5–2 and Figure 5–6 in order for the IVP program to work. The “triple name” (which is the combination of the **Server Name**, **Server Service Class**, and the **Server Service**) are imperative for the association of the IVP program to Shadow during server registry.

- **JCL DSN or Console Command**



**Notes:**

- If you are submitting the service as a batch job, specify the **JCL DSN** (see Figure 5–2).
- If you are submitting the service as started task, specify the **Console Command** (see Figure 5–6).

8. Press ENTER to perform the extract. If the extract was successful, the system will display the message “Service is now defined.”
9. Use the **END** command (or press F3) to return to the **Shadow Mainframe Adapter Server ACI Facility** options menu (Figure 5–3).

```

----- Shadow Mainframe Adapter Server ACI Facility -----
---- Subsystem SDBB
OPTION ===>

 1 ACI Server Definition - Create ACI Server Map Information
 2 Natural Extract       - Create Natural Map Information
 3 ACI Map Display       - Display ACI Server Map Information
 4 Active Server Display - Display Active ACI Servers
 5 Map Refresh           - Refresh Shadow Maps
 6 ACI Error Create      - Create ACI Error Processing Definitions
 7 ACI Error Display     - Display ACI Error Processing Definitions

```

**Figure 5–7. Shadow Mainframe Adapter Server ACI Facility Options Menu**

10. From this menu, select Option 5, Map Refresh.
11. Press ENTER to refresh the service definition into Shadow.
12. From this menu, select Option 3, ACI Map Display.

13. Press ENTER to display the member name (in this example, ACIBNIVP) as a structure name.

14. Within your Java program, issue the following SQL statement:

```
CALL SHADOW_ACI( 'SEND' , 'ACIBNIVP' , 'DATA' )
```



*Doc Reference:*

For more information about performing a query, see the *Shadow Mainframe Adapter Client for Natural Client Installation and Administration Guide*.