IBM

Shadow Mainframe Adapter Client for IMS/TM

Shadow Mainframe Adapter Server Installation
Shadow Interface for IMS/TM 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 IMS/TM.

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

NEON
SYSTEMS, INC.

Contents

Part I: Introduction

Part II: Shadow Mainframe Adapter Server Installation

Part III: Shadow Mainframe Adapter Client for IMS/TM

About this Publication

This book contains planning and installation for the Shadow Mainframe Adapter Client for IMS/TM, a licensed add-on component of the Shadow product.

How this Publication Is Organized

This book contains the following chapters:

Part I: Introduction to Shadow Mainframe Adapter Client for IMS/TM

- Chapter 1, “Introduction: Shadow Mainframe Adapter Client for IMS/TM,” which gives a brief overview of the Shadow access to IMS/TM, including the Shadow Mainframe Adapter Server, the Shadow Mainframe Adapter Client and the Shadow Interface for IMS/TM.

Part II: Shadow Mainframe Adapter Server Installation

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

Part III: Shadow Interface for IMS/TM

- Chapter 5, “Shadow Interface for IMS/TM: Installation,” decribes the planning considerations and installation steps for the Shadow Interface™ for IMS/TM.

NEON Systems, Inc. Products and Publications

For a comprehensive list of the products currently marketed by NEON Systems, Inc. (NEON), visit the following website:

`http://www.neonsys.com`

You can access and download all of the current NEON publications by navigating within this website to the following location:

`http://www.neonsys.com/downloads/documentation/default.asp`

In addition, the current NEON publications are also available on the Product Documentation CD.

Reader’s Comments

Please e-mail any comments or questions you have about our documentation to `support@neonsys.com`.

Thank you!

Part I

Introduction

CHAPTER 1: Introduction: Shadow Mainframe Adapter Client for IMS/TM

This chapter gives a general introduction to the Shadow Mainframe Adapter Client for IMS/TM, a component of the Shadow product.

Topics include the following:

- Overview
  - Shadow
  - Shadow Mainframe Adapter Client for IMS/TM

Overview

Shadow

Organizations that view investment in integration software on a project-by-project basis can license Shadow via its Shadow packaging option. Shadow focuses on the Application Server or Integration Server connectivity requirement and provides cost-effective licensing options that fit the project model perfectly.

Shadow is an efficient, easy-to-use, 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 IMS/TM consists of:

- Shadow Mainframe Adapter Server
- Shadow Mainframe Adapter Client
- Shadow Interface for IMS/TM

Key features of Shadow include:

On the Server Side:

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

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

On the Client Side:

- Provides Connect applications with an ODBC, JDBC, and J2CA API.
- Performs data and SQL dialect conversations, dynamic-to-static SQL conversions, data compression, and network optimization in conjunction with the Shadow Mainframe Adapter Server.

Shadow Mainframe Adapter Client for IMS/TM

The Shadow Mainframe Adapter Client for IMS/TM provides the necessary tools to access IMS transactions through Shadow Mainframe Adapter Server Support for IMS/APPC. IMS/TM connectivity is via the NEON Stored Procedure (Metadata Support).

This procedure is a pseudo-DB2 procedure that contains the metadata for input and output fields, as well as other required parameters for accessing IMS transactions. It stores this information in the DB2 catalog.

Note:
If metadata is required, then DB2 is required.

The NEON Stored Procedure provides a simpler, more flexible call for clients such as Crystal Reports and Power Builder. If the user needs to supply complex input data types, this method allows the input to be mapped (via the Shadow Data Mapping Facility) to pre-extracted definitions. Thus, the full range of mainframe high-level language data types can be supported, such as small integer, large integer, packed decimal, and floating point.

The Clientor enables Java applications to integrate z/OS data and transactional sources through the JDBC API. Shadow Mainframe Adapter Client is configurable and takes advantage of Java capabilities including multi-threading, connection pooling, and batch updates. Shadow Mainframe Adapter Client is JDBC 2.0 compliant and supports JDK 1.1.x, JDK 1.2.3 (J2EE) and Java servlets. It runs on a growing range of platforms including HP-UX, Sun Solaris, IBM AIX, SCO Unix, USS, Linux, and Windows..

Part II

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 Shadow.

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 (3390)	DIR 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 (3390)	DIR BLKS
34	NEON.AHTML	Shadow Web Server Auto HTML	PO	VB	19036	2	2
35	NEON.CICSLOAD	CICS load library for users of CICS-related features	PO	U	0	3	50
36	NEON.CHGEXEC	Shadow Event Publisher sample publication rule	PO	VB	1	1	1
37	NEON.CHGEXECW	Shadow Event Publisher sample publication rule	PO	VB	1	1	1
38	NEON.HTXLIB	Shadow Event Publisher sample 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=xxxxxx,
//             LABEL=(1,SL,EXPDT=98000)
//DSKCNTL      DD   DSN=prefix.CNTL,DISP=(NEW,CATLG),
//             UNIT=SYSDA,VOL=SER=xxxxxx,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=xxxxxx:` The distribution tape you received contains a serial number in the format `xxxxxx` 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 DSNAME(NEON.SV040800.LOAD) VOLUME(xxxxxx)
```

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 `SDBxIN00` 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[a]	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'060200000000000000000300',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 REUSETHREADS
- 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(DEFAULTDB2SUBSYS)   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 REUSETHREADS

The REUSETHREADS 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 REUSETHREADS, the server side reusable connections, by setting the following started task parameters:

```
“MODIFY PARM NAME(REUSETHREADS)        VALUE(YES)”
“MODIFY PARM NAME(TARGETTHREADCOUNT)   VALUE(500)”
“MODIFY PARM NAME(THREADTIMEOUT)       VALUE(300)”
```

Where:

- **REUSETHREADS** 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 INTCP.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(ITCSUBSYSTEM) 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 IMS/TM Mainframe Adapter Client Installation and Administration Guide.*

CHAPTER 3:

Shadow Mainframe Adapter Server: Authorizing Access To Resources

This chapter covers the required 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 *Shadow Mainframe Adapter Client for ADABAS: Shadow Mainframe Adapter Server Administration and Shadow Interface for ADABAS Administration* guide.
ADAxxxxx.FILyyyyy	Access to ADABAS file ID number. See the *Shadow Mainframe Adapter Client for ADABAS: Shadow Mainframe Adapter Server Administration and Shadow Interface for ADABAS Administration* 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: • SQL information. • An uncensored view of the wrap-around trace. • The underlying binary trace records. 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)*

Shadow Mainframe Adapter Server Facility	Suggested User	Resources	Access Required
Using the SDB command.	DBA, Program Products, VTAM, Operations	CONTROLBLOCK	READ
Defining links using the **ADDRESS SDB DEFINE LINK** command.	DBA, Program Products, VTAM, Operations	SDB	ADD/ALTER
Viewing links using either ISPF/SDF Option 1 or the **ADDRESS SDB DISPLAY LINK** command.	DBA, Program Products, VTAM, Operations	LINKS	READ
Modifying links using either ISPF/SDF Option 1 or the **ADDRESS SDB MODIFY LINK** command.	DBA, Program Products, VTAM, Operations	LINKS, SDB	UPDATE
Defining databases using the **ADDRESS SDB DEFINE DATABASE** command.	DBA, Program Products	LINKS, SDB	ADD/ALTER
Viewing databases using either ISPF/SDF Option 2 or the **ADDRESS SDB DISPLAY DATABASE** command.	DBA, Program Products, VTAM, Operations	DATABASES	READ
Modifying databases using either ISPF/SDF Option 2 or the **ADDRESS SDB MODIFY DATABASE** command.	DBA, Program Products	DATABASES, SDB	UPDATE
Viewing attached users using either ISPF/SDF Option 3 or the **ADDRESS SDB DISPLAY ATTACHED** command.	DBA, Program Products, VTAM, Operations	DATABASES, SDB	READ
Viewing remote users using either ISPF/SDF Option 4 or the **ADDRESS SDB DISPLAY REMOTE** 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 **ADDRESS SDB DISPLAY PARM** command.	DBA, Program Products, VTAM, Operations	USERS, SDB	READ
Modifying product started task parameters using either ISPF/SDF Option 5.2 or the **ADDRESS SDB MODIFY PARM** 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; 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(ICHRRCDE)
      ORDER   Rxxy
      ORDER   *** Previous user-defined classes ***
      ORDER   *** Previous user-defined classes ***
      ORDER   ICHRRCDE
      NAME    ICHRRCDE(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(RACFDFN) 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	SDLEPGLI	SDSLUTCK
SDBVBFB	SDLESVRU	SDSLUTDE
SDBX	SDLINK	SDSLUTKY
SDBXCOMP	SDNTLDMD	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 PKTDATA 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 PKTDATA 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:Mainframe Adapter Client

Shadow Mainframe Adapter Server: Connecting to TSO

This chapter covers the required 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:

```
ISPLLIBNEON.SV040800.LOAD

ISPMLIBNEON.SV040800.NEONMLIB

ISPPLIBNEON.SV040800.NEONPLIB

ISPTLIBNEON.SV040800.NEONTLIB

SYSEXECNEON.SV040800.EXEC(FB) (FB if using FB datasets)

SDBTRACE(Optional dd statement, see #2 below)

SDBRPCLB (Optional dd statement, see #6 below)
```

> ***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 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.

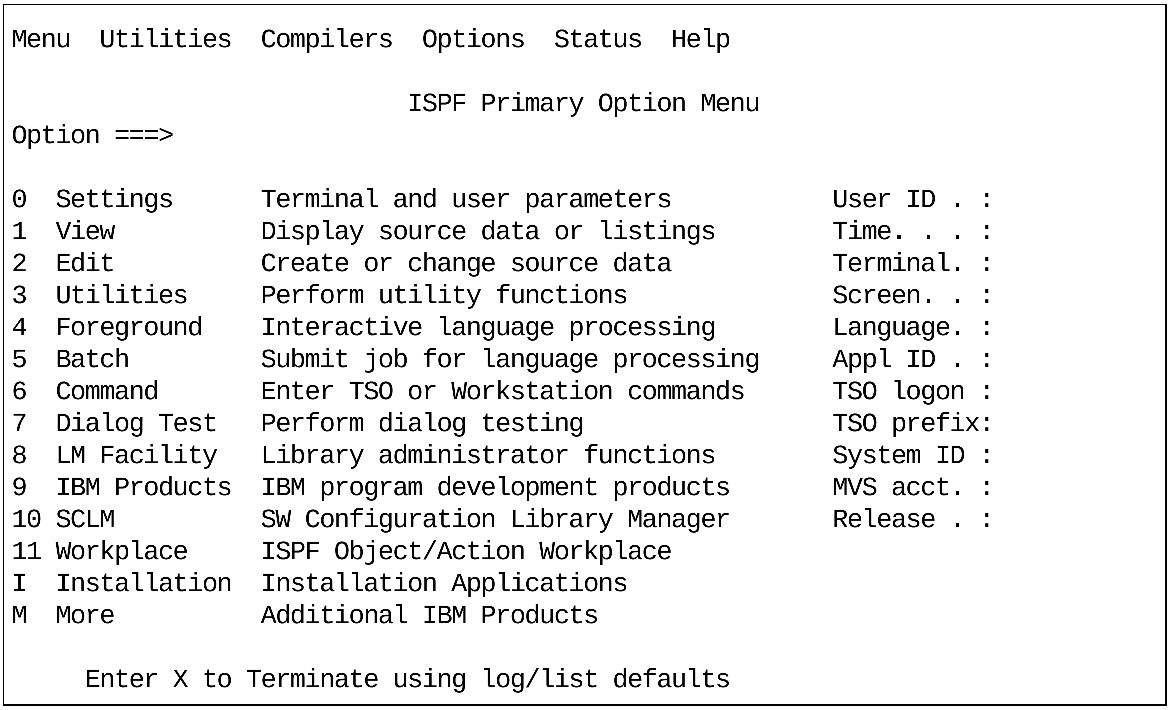
4. If you are using TCP/IP, define a new port number in the initialization EXEC for TCP/IP connections.
5. If you are using LU6.2, define and use a new APPLID.
6. To run RPCs, allocate your RPC load library to a ddname of SDBRPCLB.

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.

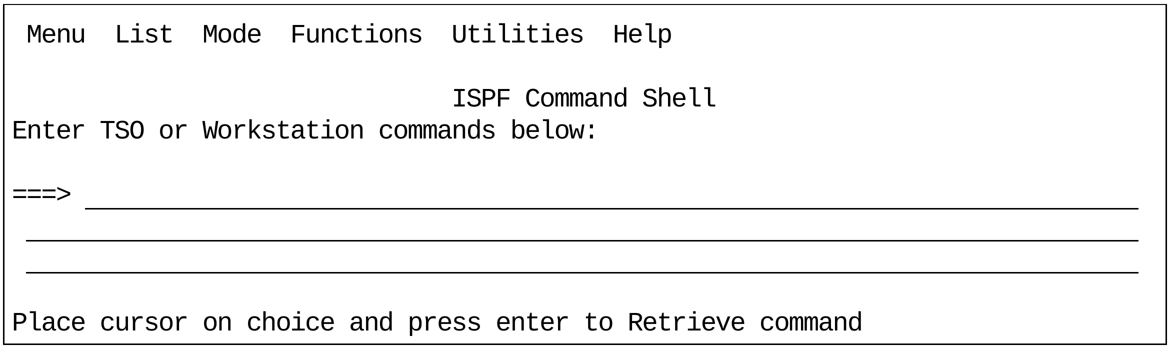


```
Menu  Utilities  Compilers  Options  Status  Help

                           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
```

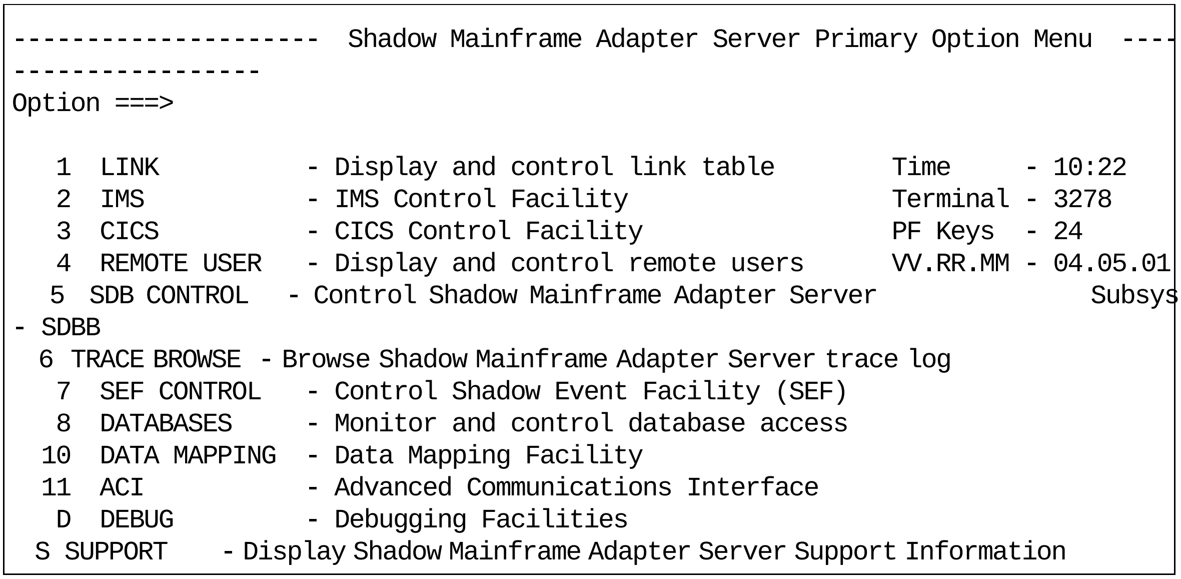
Figure 4–2. ISPF Command Shell

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 Mainframe Adapter 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.

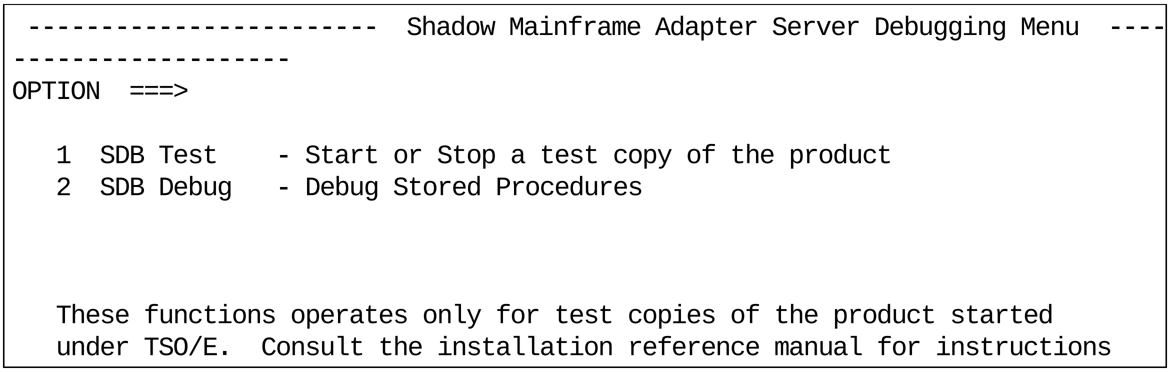


```
--------------------  Shadow Mainframe Adapter Server Primary Option Menu  ----
-----------------
Option ===>

   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 Information
```

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.

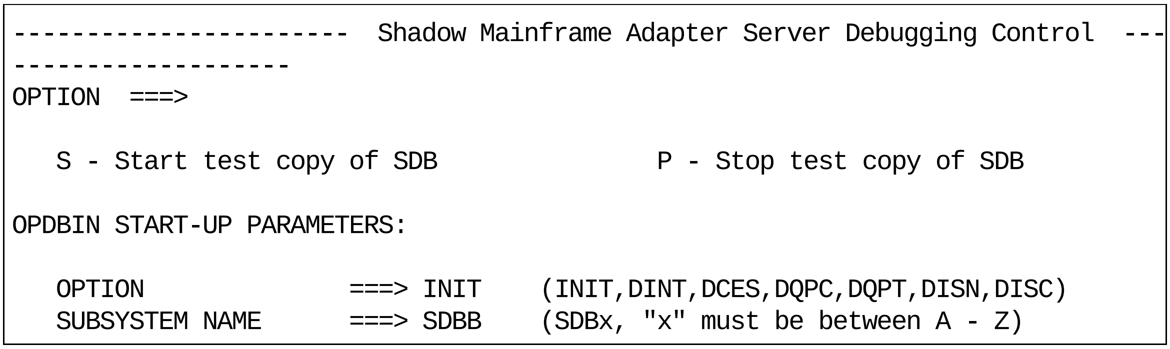


```
 -----------------------  Shadow Mainframe Adapter Server Debugging Menu  ----
-------------------
OPTION  ===>

   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
    ```

Part III

Shadow Mainframe Adapter Client for IMS/TM

CHAPTER 5:
Shadow Interface for IMS/TM: Installation

This chapter describes planning considerations for the Shadow Interface™ for IMS/TM, a licensed add-on component of the Shadow product.

Topics include the following:

- Shadow Mainframe Adapter Server Support for IMS/APPC
  - Installation Prerequisites
  - Installing the Shadow Mainframe Adapter Server Support for IMS/APPC
- Shadow Mainframe Adapter Server Support for IMS/OTMA
  - Installation Prerequisites
  - Installing the Shadow Mainframe Adapter Server Support for IMS/ OTMA

Shadow Mainframe Adapter Server Support for IMS/ APPC

Installation Prerequisites

- The Shadow Mainframe Adapter Server component has been installed.
- The Shadow Mainframe Adapter Client component has been installed.
- The Shadow Mainframe Adapter Client component has been configured and connected to the data source.
- Shadow has been licensed for the Shadow Interface for IMS/TM.
- The Shadow Mainframe Adapter Server Support for IMS/APPC has been installed.
- You are using IMS 4.1 or above.
- The MVS/APPC definitions have been set up.
- The VTAM definitions have been provided.
- The RRS/MVS is installed and running.
- The SYSTEM parameter in the IMSCTRL macro specifies an MVS version greater than or equal to 4.2. The following code provides an example:

```
IMSCTRL MACRO –
          IMSCTRL  SYSTEM=(VS/2,(ALL,DB/DC),5.1),     X
          IRLM=YES,                                    X
          IRLMNM=IRLM,                                 X
          CMDCHAR=>,                                   X
          DBRC=(YES,YES),                              X
          DBRCNM=IVP41RC1,                             X
          DLINM=IVP41DL1,                              X
```

```
DCLWA=YES,                                           X
IMSID=IVP1,                                          X
NAMECHK=(YES,S1),                                    X
MAXIO=(,015),                                        X
MAXREGN=(005,512K,A,A),                              X
MCS=(2,7),                                           X
DESC=7,                                              X
MAXCLAS=016
```

Installing the Shadow Mainframe Adapter Server Support for IMS/APPC

The installation steps for implementing IMS/APPC support include the following:

1. Modify the Shadow initialization EXEC.
2. Modify IMS to activate IMS/APPC.
3. Create VTAM APPLIDs.
4. Modify APPC/MVS.
5. **(Optional)** Install the IMS LU6.2 user edit exit.
6. Verify the installation.

Step 1: Modify the Shadow Initialization EXEC

Note:
Set the following parameters, using the **MODIFY PARM** command within the Shadow Mainframe Adapter Server initialization EXEC, `SDBxIN00`.

1. Enable the IMS Transaction Mainframe Adapter Server by setting the following parameter:

   ```
   MODIFY PARM NAME(APPC/IMS)          VALUE(YES)
   ```

2. **(Optional)** Activate the DFSLUEE0 exit by setting the following parameter:

   ```
   MODIFY PARM NAME(IMSLUEE0)          VALUE(YES)
   ```

3. **(Optional)** Set the following optional parameters using the default settings:

   ```
   MODIFY PARM NAME(IMSSYMDEST)        VALUE()
   MODIFY PARM NAME(IMSMODENAME)       VALUE(IMSLU62)
   MODIFY PARM NAME(IMSPARTNERLU)      VALUE(#BATCH)
   MODIFY PARM NAME(IMSSECURITYTYPE)   VALUE(NONE)
   ```

   Where:

   IMSSYMDEST
   Specifies a symbolic name representing the partner LU, the partner's TP name, and the mode name for the session on which the conversation is to be carried. The symbolic destination name must

match that of an entry in the side information dataset. The appropriate entry in the side information is retrieved and used to initialize the characteristics for the conversation.

If you specify a symbolic destination name, the partner LU name, mode name, and TP name are obtained from the side information. If you also specify values for the partner LU name, mode name, or TP name parameters on the allocate service, these values override any obtained from the side information.

The symbolic destination name in this field can be from 1 to 8 characters long, with characters from character set 01134. If the symbolic destination name is shorter than 8 characters, it must be left-justified in the variable field and padded on the right with blanks. If you do not want to specific a symbolic destination name, set the symbolic destination name parameter value to 8 blanks and provide values for the partner LU name, mode name, and TP name parameters.

IMSMODENAME

Specifies the mode name designating the network properties for the session to be allocated for the conversation. The network properties include, for example, the class of service to be used. The mode name value of SNASVCMG is reserved for use by APPC/MVS. If a mode name of SNASVCMG is specified on the allocate service, the request is rejected with a return code of parameter_error.

If you specify a symbolic destination name in the symbolic destination name parameter, set mode name to blanks to obtain the mode name from the side information.

If the partner LU is the same or on the same system as the local LU, mode name is ignored. If the partner LU is on a different system and you do not specify a symbolic destination name, a blank mode name defaults to any mode in effect for the local and partner LUs or causes a return code of parameter_error.

IMSPARTNERLU

Specifies the name of the IMS LU, which will be defined in SYS1.PARMLIB(APPCPMxx), as described in “Step 4: Modify MVS/APPC” on page 5-6.

IMSSECURITYTYPE

Specifies the type of security used. Possible values are as follows:

- **NONE:** Omit access security information on this allocation request.
- **SAME:** Use the userid and security profile (if present) from the allocation request that initiated the local program. The password (if present) is not used; instead, the userid is indicated as being

already verified. If the allocation request that initiated execution of the local program contained no access security information, then access security information is omitted on this allocation request.

- **PROGRAM:** Use the access security information that the local program provides on the call. The local program provides the information by means of the userid, password, and profile parameters. These values are passed exactly as specified, without folding to uppercase.

Notes:

- If IMSPARTNERLU and IMSSECURITYTYPE are not explicitly defined in the call, the system will override any values with the default.
- Do ***not*** modify the IMSCONVTYPE, INSRETURNCONTROL, or IMSSYNCLEVEL parameters unless instructed by Customer Support.

Step 2: Modify IMS to Activate IMS/APPC

1. Issue the **/START APPC** command to start APPC without restarting IMS.
2. In the startup definitions for your IMS DCCTL region, set APPC=YES. Depending on your installation, this could be member DFSPBxxx in your IMS PROCLIB dataset.

Doc Reference:

For additional details, see the *IMS Data Communication Administration Guide*.

IMS/APPC Security

IMS/APPC security is handled through the following RACF resource classes:

- **TIMS:** For IMS transaction programs.
- **CIMS:** For IMS commands.

Under ACF2 and Top Secret, if these resource classes are not defined, you will receive an ATB_SECURITY_NOT_VALID message if attempting to connect to IMS via the IMS Transaction Mainframe Adapter Server. Under ACF2 and Top Secret, the default is no access if a resource is not defined.

After verifying that these rules have been set up, you can experiment with setting IMS/APPC security to FULL or CHECK to see which one works for your environment. Both FULL and CHECK will provide the same level of security. IMS/APPC security can be changed via the IMS **/SECURE** command, as follows:

```
/SECURE APPC CHECK
```

Step 3: Create VTAM APPLIDs

The following VTAM APPLIDs can be defined:

- **(Required)** Base APPC APPL
- **(Optional)** Local APPC APPL
- **(Required)** IMS APPC APPL

> ***Note:***
> The base APPC APPL may have already been defined if you have other APPC applications.

1. Define the VTAM APPLIDs using the following:

```
MVSLU01 APPL ACBNAME=MVSLU01,BASE ACBNAME FOR APPC+
   APPC=YES,+
   AUTOSES=0,+
   DDRAINL=NALLOW,+
   DLOGMOD=#BATCH,+
   DMINWNL=5,+
   DMINWNR=5,+
   DRESPL=NALLOW,+
   DSESLIM=10,+
   LMDENT=19,+
   PARSESS=YES,+
   SECACPT=AVPV,+
   SRBEXIT=YES,+
   VPACING=0+

MVSLU02 APPL   ACBNAME=MVSLU02,LOCAL ACBNAME FOR APPC+
   APPC=YES,+
   AUTOSES=0,+
   DDRAINL=NALLOW,+
   DLOGMOD=#BATCH,+
   DMINWNL=5,+
   DMINWNR=5,+
   DRESPL=NALLOW,+
   DSESLIM=10,+
   LMDENT=19,+
   PARSESS=YES,+
   SECACPT=AVPV,+
   SRBEXIT=YES,+
   VPACING=0+

IMSLU62 APPL ACBNAME=IMSLU62,BASE ACBNAME FOR APPC/IMS+
   APPC=YES,+
   AUTOSES=0,+
   DDRAINL=NALLOW,+
   DLOGMOD=#BATCH,+
   DMINWNL=5,+
   DMINWNR=5,+
   DRESPL=NALLOW,+
   DSESLIM=10,+
   LMDENT=19,+
   PARSESS=YES,+
   SECACPT=AVPV,+
   SRBEXIT=YES,+
   VPACING=0
```

2. Activate the defined APPLIDs using the following:

   ```
   V NET,ID=XXXXX,ACT
   ```

Step 4: Modify MVS/APPC

Define the IMS LU and the additional local LU to APPC in SYS1.PARMLIB(APPCPMxx) as follows:

```
/************************************************************/
/*                                                          */
/* LIB: SYS1.PARMLIB(APPCPM00)                              */
/* GDE: CBIPO MVS INSTALLATION                              */
/* DOC: THIS PARMLIB MEMBER DEFINES A LU TO APPC, ALONG     */
/*     WITH A VSAM DATASET FOR TP PROFILES AND A SECOND     */
/*     ONE FOR SIDE INFORMATION                             */
/*                                                          */
/*      THE APPC PARMLIB MEMBER IS SPECIFIED ON THE         */
/*      START AND SET OPERATOR COMMANDS                     */
/*                                                          */
/*      THIS PARMLIB STATEMENT IS DESIGNED TO SUPPORT       */
/*      SAMPLES IN SYS1.SAMPLIB. REFER TO SYS1              */
/*      TO SYS1.SAMPLIB(ATBALL) FOR A LIST OF SUPPLIED      */
/*      SUPPLIED SAMPLE MATERIALS.                          */
/*                                                          */
/************************************************************/

LUADD ACBNAME(MVSLU01) BASE TPDATA(SYS1.APPCTP)
   SIDEINFO DATASET(SYS1.APPCSI)
LUADD ACBNAME(IMSLU62) SCHED(IVP1) BASE TPDATA(SYS1.APPCTP)
   TPLEVEL(SYSTEM)
LUADD ACBNAME(MVSLU02) NOSCHED TPDATA(SYS1.APPCTP)
```

Note:

This can be added dynamically by putting the define statements in SYS1.PARMLIB member APPCPMxx and issuing the **SET APPC=xx** command (T APPC=xx).

Step 5: (Optional) Install the IMS LU 6.2 User Edit Exit (DFSLUEE0)

When an IMS transaction starts, it has the ability to determine that another transaction code must process the user request. This results in an IMS transaction message switch. Ultimately, the output message is in a different format than originally defined by the NXT specification in the MFS source.

The information required to properly format the output message must be communicated to the Shadow Mainframe Adapter Server. In order to avoid non-standard interfaces to IMS, this information must be obtained through a programmable and documented interface, the IMS LU 6.2 user edit exit (DFSLUEE0). This load module can be found in the load library hlq.LOAD(DFSLUEE0).

Install the IMS LU 6.2 user edit exit (DFSLUEE0) as follows:

1. Relink this exit into your IMS RESLIB, which can be found in the sample library hlq.SAMP(JCLLUEE0).

2. Stop and restart your IMS system for the exit to take effect.

Step 6: Verify the Installation

Perform a query to verify installation.

Doc Reference:
For more information about performing a query, see the *Shadow Mainframe Adapter Client for IMS/TM Mainframe Adapter Client Installation and Administration Guide.*

Shadow Mainframe Adapter Server Support for IMS/OTMA

Installation Prerequisites

Make sure of the following:

Make sure of the following

- You are using IMS 5.1 or above.
- IMS OTMA use requires an OS/390 or z/OS coupling facility (XCF) to be in use by the OS/390 or z/OS image containing the IMS subsystem being used. The XCF must be active in either monoplex or sysplex mode.
- The RRS/MVS is installed and running.

Installing the Shadow Mainframe Adapter Server Support for IMS/OTMA

The installation steps for implementing IMS support using IMS/OTMA include the following:

1. Modify IMS to enable IMS/OTMA.
2. Define IMS/OTMA global options in the initialization EXEC.
3. Define IMS/OTMA connections in the initialization EXEC.
4. Verify IMS/OTMA installation.
5. Determine IMS/OTMA security considerations.
6. Verify Shadow Mainframe Adapter Server Support for IMS/OTMA installation.

Step 1: Modify IMS to Enable IMS/OTMA

IMS/OTMA support must be enabled in each IMS system that uses Shadow IMS/OTMA services. The *IMS/ESA Open Transaction Manager Access Guide and Reference* manual contains information regarding installation of OTMA support within an IMS system.

The IMS/OTMA installation steps necessary to allow use of Shadow Mainframe Adapter Server Support for IMS/OTMA are as follows:

1. Add the OTMA= parameter to member DFSPBxxx in IMS PROCLIB to specify the OTMA enable option. Possible values are as follows:

   - **Y:** IMS will enable OTMA support during IMS initialization.
   - **N:** IMS will not enable OTMA during IMS initialization, but OTMA can be enabled using the **/START OTMA** command if GRNAME was properly defined as described below.

2. Add the GRNAME= parameter to member DFSPBxxx in IMS PROCLIB to specify the Cross System Coupling Facility (XCF) group name that IMS should use. The group name you select must be alphanumeric and 1 to 8characters in length. IMS joins the XCF group name specified when OTMA is started during IMS initialization or when the **/START OTMA** command is issued. Each IMS subsystem using OTMA must be supplied a unique GRNAME value. You are not required to define XCF definitions in relation to the GRNAME value.

3. Add the OTMANM= parameter to member DFSPBxxx in IMS PROCLIB to specify the XCF member name that IMS will use when it joins the IMS/OTMA XCF group. The member name selected must be alphanumeric and 1 to 16 characters in length. You are not required to define any XCF definitions in relation to the OTMANM value. This parameter is optional. When omitted, it will default to the IMS APPLID name present on the APPLID1= parameter also contained in the member DFSPBxxx in IMS PROCLIB.

Example

An example of this definition follows:

```
OTMA=Y,
GRNAME=IMS1OTMG,
OTMANM=IMS1OTMA,
```

The above definitions would cause IMS/OTMA to be initialized when IMS is started, the OTMA XCF group name used for the IMS subsystem would be IMS1OTMG, and the OTMA XCF member name used by IMS when joining the group would be IMS1OTMA.

Step 2: Define IMS/OTMA Global Options in the Initialization EXEC

To define the IMS/OTMA global options, modify the following parameters in the initialization EXEC using the **MODIFY PARM** command within the Shadow Mainframe Adapter Server initialization EXEC, `SDBxIN00`:

```
IF 1 = 1 THEN
 DO
MODIFY PARM NAME(IMSOTMA)                 VALUE(YES)
MODIFY PARM NAME(IMSOTMADEFCON)           VALUE(IMSTEST1)
MODIFY PARM NAME(IMSOTMADEFSEC)           VALUE(SAME)
MODIFY PARM NAME(IMSOTMADEFMAP)           VALUE(MAPNAME)
MODIFY PARM NAME(TRACEOTMAEVENTS)         VALUE(YES)
MODIFY PARM NAME(TRACEOTMADETAIL)         VALUE(NO)
MODIFY PARM NAME(TRACEOTMABUFFERDATA)     VALUE(NO)
MODIFY PARM NAME(IMSOTMAEXECIMS)          VALUE(YES)
 END
```

Where:

IMSOTMA

Is the IMS/OTMA support enable option. The possible values are YES and NO. Specify YES to enable Shadow Mainframe Adapter Server Support for IMS/OTMA. The default value is NO.

IMSOTMADEFCON

Specifies the default Shadow IMS/OTMA connection name to be used if the Shadow OTMA connection name is omitted on an OTMA-related request. Definition of connection names is discussed in the following section. The default is none, so no default connection name is used.

IMSOTMADEFSEC

Specifies the default Shadow IMS/OTMA transaction security type. The security type determines what authorization ID if any is passed to IMS for transaction level security authorization. Possible values are as follows:

- **NONE:** Omits any security information on the request when the transaction is passed to IMS/OTMA.
- **SAME:** Specifies use of the userid and security profile information that exists from the client connection that initiated the OTMA request. The userid and profile, if present, will be passed to IMS when the transaction is initiated.
- **PROGRAM:** Specifies use of the access security information that the local program provides in the OTMA request API parameters. The local program provides the information through the USERID and PROFILE parameters. These values are passed as specified to IMS/OTMA without modification.

IMSOTMADEFMAP
Specifies the default Shadow IMS/OTMA map name.

TRACEOTMAEVENTS
Determines if high level tracing will occur for IMS/OTMA event activity. Specify YES. Valid options are YES or NO.

TRACEOTMADETAIL
Determines if detail level tracing will occur for IMS/OTMA event activity. Specify NO. Valid options are YES or NO.

TRACEOTMABUFFERDATA
Determines if buffer data level tracing will occur for IMS/OTMA event activity. Specify NO. Valid options are YES or NO.

IMSOTMAEXECIMS
Determines whether all /*EXECIMS activity will use native OTMA. When set to YES, all /*EXECIMS activity to use native OTMA. If you want to use one IMS subsystem to flow through EXECIMS using APPC and another using OTMA, set this value to NO and define keywords within the rules to indicate the OTMA use you want.

Step 3: Define IMS/OTMA Connections in the Initialization EXEC

To define IMS/OTMA connections in the initialization EXEC, set the following parameters using the **DEFINE IMSOTMA** command in the SDBxIN00:

```
IF 1 = 1 THEN
 DO
"DEFINE IMSOTMA CONNID(IMSTEST1) IMSID(IMS1)",
        "IMSXCFGRP(IMS1OTMG) IMSXCFMEM(IMS1OTMA)",
        "TPIPEMIN(05)       TPIPEMAX(10)",
        "RESPTIMEOUT(120)   CONNTIMEOUT(300)",
        "CONVAPPC(YES)      CONVAPPCLU(IMS1APLU)"
 END
```

Where:

CONNID

(Required) Specifies the Shadow IMS/OTMA connection name assigned to the Shadow IMS/OTMA connection for the selected IMS subsystem name. The connection name is used as the identifier of the IMS/OTMA connection and is used by Shadow IMS/OTMA requestors to identify which IMS subsystem they want to communicate with when passing API requests to the Shadow address space. It provides a level of abstraction so the IMS ID can be changed without having to modify or recompile requesting applications.

A default IMS connection name can be specified using the IMSOTMADEFCON global keyword discussed in "Step 2: Define IMS/OTMA Global Options in the Initialization EXEC" on page 5-9. This allows requestors to omit the Shadow IMS/OTMA connection name within API requests and to use the connection name specified in the IMSOTMADEFCON parameter. Multiple Shadow IMS/OTMA connections can be defined and used from within a single Shadow Mainframe Adapter Server address space. The connection name is alphanumeric and one to eight characters in length. The name must be unique for each Shadow IMS/OTMA connection defined within a Shadow Mainframe Adapter Server address space.

IMSID

(Required) Specifies the IMS ID of the IMS subsystem that is the target of the Shadow IMS/OTMA connection.

IMSXCFGRP

(Required) Specifies the IMS/OTMA XCF group name in use for the IMS subsystem. This name is defined for the IMS subsystem in the DFSPBxxx IMS PROCLIB member. The keyword name is GRNAME=. Set this parameter to the contents of the GRNAME= parameter in use by the IMS subsystem, as defined in "Step 1: Modify IMS to Enable IMS/OTMA" on page 5-8.

IMSXCFMEM

(Required) Specifies the IMS/OTMA XCF member name in use for the IMS subsystem. This name is defined for the IMS subsystem in the DFSPBxxx IMS PROCLIB member, keyword name OTMANM=. Set this parameter to the contents of the OTMANM= parameter (see "Step 1: Modify IMS to Enable IMS/OTMA" on page 5-8) or the APPLID1= parameter in use by the IMS subsystem. If not defined, it defaults to the IMS APPLID specified in the APPLID1= parameter.

TPIPEMIN

(Optional) Specifies the minimum number of OTMA transaction pipes the Shadow Mainframe Adapter Server address space will use to service Shadow IMS/OTMA request activity. Shadow IMS/OTMA server shares OTMA transaction pipes across multiple IMS/OTMA users. The default is 3, and the valid range in 1 to 100.

TPIPEMAX

(Optional) Specifies the maximum number of OTMA transaction pipes the Shadow Mainframe Adapter Server address space will use to service Shadow IMS/OTMA request activity. Shadow IMS/OTMA server dynamically increases the number of OTMA transaction pipes used up to this value if warranted by application workload activity. Shadow IMS/OTMA server shares OTMA transaction pipes across multiple IMS/OTMA users. The default is 5, and the valid range is 1 to 100. If value specified is less than TPIPEMIN, it will be set to the TPIPEMIN value in use.

RESPTIMEOUT

Is the Shadow IMS/OTMA response time-out value. It is expressed in seconds. The valid range is 0 to 900. If you specify 0, response time-out is disabled for the connection. Use of this operand enables application requests that have not received a response for IMS/OTMA for a specified period of time to be timed-out, rather than leaving a client hung awaiting a response. The default value is 120 seconds. This parameter is optional.

CONNTIMEOUT

(Optional) Specifies the Shadow IMS/OTMA connection time-out value (in seconds). If you specify 0, connection time-out is disabled for the connection. Use of this operand enables Shadow IMS/OTMA users to be terminated if the user's connection has had no request activity for the specified period of time. The default value is 300 seconds, and the valid range is 0 to 300.

CONVAPPC

(Optional) Allows existing Shadow IMS/APPC applications to be processed using IMS/OTMA rather than IMS/APPC. Possible values are as follows:

- **NO:** (Default) Continue processing Shadow IMS/APPC applications using IMS/APPC.
- **YES:** Enable all Shadow IMS/APPC request activity to dynamically execute using IMS/OTMA instead of IMS/APPC. Use of this option requires no changes to existing Shadow IMS/APPC applications. In addition, Shadow IMS/APPC does not have to be active in the Shadow Mainframe Adapter Server address space. If YES is specified, the CONVAPPCLU parameter must also be specified.

CONVAPPCLU

(Required if CONVAPPC is set to YES) Specifies the IMS/APPC LU name of the IMS subsystem for which this connection is being defined.

Step 4: Verify IMS/OTMA Installation

If OTMA was enabled in your IMS subsystem(s) using the procedures described above, you can verify IMS/OTMA enablement as described below. If OTMA was already enabled and functional in your IMS subsystem(s), you can skip this step.

Verifying that IMS/OTMA Has Been Enabled

1. Start the IMS subsystem using normal operating procedures.

2. View the IMS control region joblog. The following message should be present in the initialization message flow.

   ```
   DFS2360I 09:52:12 XCF GROUP JOINED SUCCESSFULLY. IMS1
   ```

3. If the message is present, OTMA was successfully enabled. If not, then examine the joblog for IMS error messages related to OTMA initialization. If you cannot resolve the problem, contact NEON Systems Customer Support for assistance.

4. Issue the IMS **/DIS OTMA** command. The system should display the following:

   ```
   GROUP/MEMBER XCF-STATUS USER-STATUS  SECURITY IMS1OTMG
    -IMS1OTMA     ACTIVE     SERVER     FULL     *00174/131059*
   ```

The sample output shown above indicates the IMS subsystem OTMA XCF group name is IMS1OTMG, the IMS subsystem OTMA member name is IMS1OTMA, status is active, and no other members (applications) are currently active in the OTMA group.

Step 5: Determine IMS/OTMA Security Considerations

IMS/OTMA allows you to implement the following levels of security:

- Connection level security
- Transaction level security

The type of security implemented is installation dependent. The following sections describe the security options available and how they can be implemented.

It is highly recommended that minimal security needs be implemented during the initial installation and that additional security be added after installation has been completed.

Doc Reference:

Additional information related to IMS/OTMA security can be found in the *IMS/ESA OTMA Guide and Reference* and other IMS/ESA publications.

IMS/OTMA Connection Level Security

IMS/OTMA connection level security lets you control what server address spaces have the ability to join the IMS/OTMA XCF group. This level of security is implemented using the RACF FACILITY class. OTMA connection level security is optional. Use of connection level security is dependent on your installation security requirements.

The RACF FACILITY class must be active in your RACF environment to implement connection level security. To implement connection level security you must define the OTMA facility name using a required naming convention. You must also grant READ access to the facility for the Shadow Mainframe Adapter Server address space that you connect to the IMS OTMA XCF group.

Complete the following steps for each Shadow Mainframe Adapter Server address space for which you are implementing OTMA connection level security:

1. Define the RACF facility name using the required IMS/OTMA naming convention as follows:

   ```
   IMSXCF.group.member
   ```

   Where:

   group

   Specifies the IMS/OTMA XCF group name. The group name is the name specified in the IMSXCFGRP parameter in the **DEFINE IMSOTMA** connection definition for the IMS system for which you are defining the facility.

   member

   Specifies the Shadow Mainframe Adapter Server address space IMS/OTMA XCF member name. The member name is OTMA XCF member name used by the Shadow Mainframe Adapter Server address space when it joins the OTMA group. The Shadow Mainframe Adapter Server member name used is always the server subsystem name concatenated with the OS/390 or z/OS system name.

   For example, the following RACF command defines the facility for IMS/OTMA group IMS1OTMG, member SDBOSYSTEM1 (where SDBO, the Shadow Mainframe Adapter Server subsystem name, is concatenated with SYSTEM1, the OS/390 or z/OS system name on which the Shadow Mainframe Adapter Server address space executes):

   ```
   RDEFINE FACILITY IMSXCF.IMS1OTMG.SDBOSYSTEM1
   ```

2. Grant READ access to the facility for the Shadow Mainframe Adapter Server address space. This allows access to the facility that enables access to the IMS/OTMA XCF group.

For example, the following RACF command permits Shadow Mainframe Adapter Server address space SDBO access to the facility defined in the previous step:

```
PERMIT  IMSXCF.IMS1OTMG.SDBOSYSTEM1 CLASS(FACILITY) ID(SDBO)
ACCESS(READ)
```

IMS/OTMA Transaction Level Security

IMS/OTMA transaction level security lets you implement IMS transaction level security for transactions originating from a Shadow Mainframe Adapter Server address space using OTMA to execute IMS transactions. The type of OTMA transaction level security to be used is determined by your installations security requirements.

Implementing IMS transaction level security using IMS/OTMA and Shadow Mainframe Adapter Server requires setting the desired security level within IMS and setting the appropriate security settings within Shadow Mainframe Adapter Server. The following sections describe these areas in further detail.

Modifying IMS for OTMA Transaction Level Security. The type of OTMA transaction security in use by your IMS subsystem is controlled using the / **SECURE OTMA** IMS command. This command is used to set the type of IMS transaction security to use for IMS transactions originating from IMS/OTMA clients. The Shadow Mainframe Adapter Server address space is an IMS/OTMA client.

Possible security options using the **/SECURE OTMA** command are as follows:

- **NONE:** No RACF transaction level security will be used.
- **CHECK:** RACF will be used for transaction security, IMS commands are checked using the RACF CIMS resource class, IMS transactions are checked using the RACF TIMS resource class.
- **FULL:** RACF will be used for transaction security, IMS commands are checked using the RACF CIMS resource class, IMS transactions are checked using the RACF TIMS resource class. Additional RACF calls are done to create the security environment for IMS dependent regions.
- **PROFILE:** RACF will be used for transaction security. Use of PROFILE is the same as FULL with one exception. When PROFILE is used transaction security can be bypassed when the security type specified within a requestor's IMS/OTMA API parameter list is set to NONE.

Modifying Shadow Mainframe Adapter Server for OTMA Transaction Level Security. Shadow supports all IMS/OTMA security options provided by IMS as discussed in the previous section. Shadow Mainframe Adapter Server allows flexibility in selecting the authorization id to be used for execution of a IMS transaction using IMS OTMA.

The authorization id used by IMS can be the same authorization id used to establish the Shadowion or a different authorization id. Control of what authorization id is used is implemented using the IMSOTMADEFSEC Shadow Mainframe Adapter Server parameter to set the global option and the security related parameters which may specified within a Shadow IMS/OTMA request using the API parameter list security options.

Doc Reference:

For a list of the valid default security options that may be specified, refer to the IMSOTMADEFSEC parameter description on page 5-9.

Step 6: Verify Shadow Installation

If installation steps 2, 3, and 4 have been successfully completed, we are ready to start the Shadow started task and establish IMS/OTMA connections to all IMS subsystems defined using the **DEFINE IMSOTMA** parameters discussed in step 3.

Perform the following to verify Shadow Mainframe Adapter Server Support for IMS/OTMA was successfully installed and functional:

1. Start the Shadow started task.

2. View the Shadow joblog or OS/390 or z/OS syslog. Verify that the following messages are present in the initialization message flow.

```
SDO1290H SDBO     IMSOTMA              YES
SDO1290H SDBO     IMSOTMACONNID        'IMSTEST1'
SDO1290H SDBO     TRACEOTMAEVENTS      YES
SDO1290H SDBO     TRACEOTMADETAIL      NO
SDO1290H SDBO     TRACEOTMABUFFERDATA  NO
…
SDO0603I IMS/OTMA SERVER INIT IN PROGRESS FOR IMS SSID=IMS1
SDO0603I IMS/OTMA SERVER INIT IN PROGRESS FOR IMS SSID=IMS2
…
SDO0604I IMS/OTMA SERVER INIT COMPLETE FOR IMS SSID=IMS1
SDO0602I IMS/OTMA SERVER WAITING OTMA INIT FOR IMS SSID=IMS2
```

   The sample messages above indicate IMS/OTMA use was defined for two IMS subsystems, IMS1 and IMS2. The messages indicate the following:

   - Initialization (“INIT”) was successful for both IMS subsystems.
   - Shadow Mainframe Adapter Server Support for IMS/OTMA is active for subsystem IMS1.
   - Shadow Mainframe Adapter Server Support for IMS/OTMA is waiting for IMS2 to be started. When it becomes active, Shadow will establish an IMS/OTMA connection with the subsystem.

If this type of message flow is not seen, then look for error messages related to OTMA initialization failure. Attempt to resolve the failure incurred, and contact NEON Systems Customer Support if assistance is needed.

3. Issue the IMS **/DIS OTMA** command. The following output should be displayed.

```
GROUP/MEMBER      XCF-STATUS   USER-STATUS      SECURITY
IMS1OTMG
-IMS1OTMA          ACTIVE        SERVER         FULL
-SDBODEV1          ACTIVE        ACCEPT TRAFFIC
*00174/140200*
```

   Sample output provided indicates the following:

   - The IMS/OTMA XCF group name is IMS1OTMG.
   - The OTMA member name is IMS1OTMA, and the status is active
   - The Shadow started task SDBO has an active IMS/OTMA connection

4. Perform a query to verify installation.

Doc Reference:
For more information about performing a query, see the *Shadow Mainframe Adapter Client for IMS/TM Mainframe Adapter Client Installation and Administration Guide.*