SHADOW

INSTALLATION GUIDE

NEON
SYSTEMS, INC.

This document is published by the NEON Systems, Inc. Technical Publications Department and applies to Shadow Server, Version 4, Release 5, and Shadow® OS/390 Web Server™, Version 4, Release 5.

Restricted Rights Legend

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 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, Technical Support)
Fax: (281) 242-3880

December 13, 1999

Contents

Appendix A: Distribution Tape Contents

Appendix B: Providing Access to Shadow Resources

Appendix C: Shadow Logging

Appendix D: IMS Connectivity Options (Includes AutoHTML)

About this Publication

This book describes the installation process for the Shadow Server, the server-side component of both *Shadow Direct* and *Shadow OS/390 Web Server.*

How this Publication is Organized

This book contains the following chapters and appendixes:

- Chapter 1, “Installing Shadow Server,” provides details for the step-by-step installation procedure.
- Appendix A, “Distribution Tape Contents,” lists the contents of the Shadow distribution tape.
- Appendix B, “Providing Access to Shadow Resources,” provides information about defining classes and resources.
- Appendix C, “Shadow Logging,” describes the Shadow Logging facility, and the process for enabling it.
- Appendix D, “IMS Connectivity Options (Includes AutoHTML),” explains how to install and configure IMS support for *Shadow Direct* and *Shadow OS/390 Web Server.*
- Appendix E, “Transaction Server for CICS,” explains how to install and configure the CICS transaction server.
- Appendix F, “AutoHTML - Web Enabling Transactions (SWS),” covers the configuration steps to Web enable your transaction.
- Appendix F, “Configuring Secure Sockets Layer (SSL) Support,” gives complete step-by-step instructions for configure SSL support.
- Appendix G, “Link-Editing the DSN3@ATH Exit,” explains how to use the DSN3@ATH exit to propagate your userids.
- Appendix H, “Load Balancing,” describes load balancing which automatically directs inbound connections to the copy of Shadow Server with the greatest resources available.
- Appendix I, “WLM Enablement,” describes the WorkLoad Manager, a component of the OS/390 operating system.
- Appendix J, “Setting up Two Shadow Servers,” describes how to start up additional Shadow Servers.
- Appendix K, “Detailed Description of Shadow Direct Connection Modes,” discusses connection modes supported by Shadow Direct.

- Appendix L, “Installing a Maintenance Tape,” provides information about the distribution tape containing the libraries needed for the particular version of Shadow Server.
- Appendix M, “Setting up Shadow Server to Run under User’s TSO Address Space,” contains the information to setup and test Shadow Server running under a user’s TSO address space.
- Appendix N, “Shadow_VSAM for CICS,” describes how to install the Shadow_VSAM component for CICS. This component allows for the access and update of CICS assigned KSDS VSAM files.
- Appendix O, “Recoverable Resource Manager Services Attachment Facility (RRSAF) Support,” describes RRSAF and how it works, and provides guidelines as to when to use RRSAF with NEON products, and how to configure RRSAF at your site.
- Appendix P, “AutoHTML - Web Enabling Transactions (SWS),” describes the procedure for executing online IMS transactions and commands.

Conventions

This book contains the following highlighting conventions:

BOLD CAPS
Identifies commands. For example:

Use the **KEYS** command to ...

Text enclosed in single quotes denotes library, data set, and DD names.
For example:

```
‘SLDSYSIN’    ‘PLUSIN’    ‘RESLIB’
```

`Monospace`
Identifies code examples, screen prompts, and messages, as well as directory paths. For example:

```
//STEP010    EXEC  PGM=NDBA2400
```

`Monospace Italics`
Identifies information you must provide at a screen prompt or in a text field. For example:

PARM=’PARMLIB=*your.parmlib*’

<KEY> Identifies the key to press. For example:

<ENTER>

NEON Systems, Inc. uses *Release*.*Version* to identify software packages. For example, *Product 4.1*, denotes the fourth release, first revision of the software.

Reader’s Comments

At NEON Systems, Inc. we are always looking for good ideas. If you have any comments or suggestions regarding any of our publications, please complete the Reader’s Comment form (located at the back of this book) and return it to NEON, Attention: Technical Publications Department.

Mailing Address: **NEON Systems, Inc.**
14100 SW Freeway, Suite 500
Sugar Land, Texas 77478

Fax Number: (281) 242-3880

You can also send comments to directly to our Technical Publications department via the following e-mail address: **documentation@neonsys.com**.

Thank you!

NEON Systems, Inc. Products and Publications

For a comprehensive list of the products currently marketed by NEON Systems, Inc., (*NEON*) visit our World Wide Web site at: **http://www.neonsys.com**.

You can also access and download all of the current NEON publications from this Web site.

Year 2000 Compliancy Statement

The following products from NEON Systems, Inc., are Year 2000 ready:

- **Enterprise Security Management Products**
- **Enterprise Subsystem Management Product Family**
- **Shadow® Product Family and Add-On Components**

  The mainframe code for the Shadow Product Family, Version 3.1 and all subsequent versions, are Y2K ready.

  All versions of the client code associated with Shadow® Direct™ and Shadow Enterprise Direct® are Y2K ready.

  Note:

  While Shadow Direct, Shadow® OS/390 Web Server™, and Shadow Enterprise Direct are Y2K ready, customers should be aware that these products can provide access to data sources that may not be Y2K ready.

These products use four-digit year values both internally and externally (although, in a few cases, two-digit year values are displayed while four-digit year values are maintained internally).

Working with Technical Support

NEON Systems, Inc. provides a number of ways for you to obtain assistance for our products. All product support inquiries are handled by the same support group, regardless if you are a trial or a licensed customer. The following are available support options:

Support Option	How to Access	How it Works	This Option is Best for:
E-mail	To contact Technical Support via e-mail: `support@neonsys.com` Email is available for receipt 24 hours a day, 7 days a week and is answered between 9AM-7PM CST Monday through Friday.	Email goes to the support queue, which is continuously monitored by a staff of cross-functional technical experts. It is answered in the order it is received. It is logged in the support database and assigned a trouble ticket number for tracking purposes.	This type of support is excellent for low to medium priority requests. It is a proven method for providing further information on critical problems that may have been phoned in. Email is a convenient way of sending us a list of lower priority items you have collected at a time that is convenient for you.
Phone	To contact Technical Support, please call: **1-800-505-6366** (U. S. and Canada) **1-281-491-4200** (outside North America)	During normal working hours you will be transferred to someone who can usually answer your question on the first call. You may be required to page a support person via our phone mail system after hours.	This type of support is best for high priority requests and initial installation questions. Use this option for any obvious system errors or anytime you need the most rapid reply to your question.
Internet	To access Internet support, please visit our Web site at: `www.neonsys.com`	Simply visit our Web site. NEON Systems works to keep current, relevant materials on our Web site to support our trial and licensed customers.	This option provides immediate access to documentation, updated client-side drivers, and our product Knowledge Base. The Knowledge Base is a collection of questions answered by support. Use this option to answer your own questions or to get a better understanding of what customers ask on an ongoing basis.
Account Manager	To contact your NEON Systems Sales Representative, please call: **1-800-505-6366** (U. S. and Canada) **1-281-491-4200** (outside North America)	Your Sales Representative is your account manager. This person is ultimately responsible for your complete satisfaction with NEON Systems, Inc.	Contact your Sales Representative for pricing information, contract details, password renewal or if you feel your needs are not being met.

CHAPTER 1: Installing Shadow Server

This process provides the steps for installing the Shadow Server component of Shadow Direct, as well as the Shadow OS/390 Web Server. Installing both products now is the most efficient way to experience the unique benefits each has to offer. Throughout this document, the term **Shadow Server implies both products unless explicitly stated otherwise**.

Appendices provide for variations of the standard installation. This is where you will find information specific to Shadow Logging, Shadow Resources, IMS, CICS, VCF, SSL, DSN3@ATH exit, maintenance tapes, as well as the distribution tape contents.

Check your Distribution Package

Before installing Shadow Server, please take a moment to verify that you received a complete distribution package. The package should contain the following items:

- The distribution media (in tape cartridge form).
- The *Shadow Server User's Guide* and/or *Shadow OS/390 Web Server Guide.*
- The *Shadow Server Messages Guide.*
- The *Shadow Programming Guide.*

If you find that items are missing, please contact NEON Systems, Inc. before continuing.

Verify Software Levels

Shadow Server requires the following host software to operate:

- MVS/ESA (any level)
- VTAM 3.2 or later (Shadow Direct only)
- TSO/E Version 2 or later
- ISPF 2.3 or later
- DB2 Version 2 or later
- Any IBM-supported release of RACF, ACF2 Release 4.1 or later, Top Secret

Please contact NEON Systems, Inc., if you cannot meet these software-level requirements.

Installation Quick Reference (An Overview)

The following installation is the most efficient path to enjoying the benefits of NEON's products. Table 1–1 on page 1-2 will outline the installation steps for Shadow Server, and Table 1–2 on page 1-4 will outline the installation steps for Shadow OS/390 Web Server.

Shadow Server

These steps are all discussed in detail either in this chapter or in one of the appendices (see Reference column in Table 1–1 on page 1-2).

Shadow Server Installation

No.	Step	Req.	Opt.	Reference
1	Unload the CNTL dataset.	✓		See “Step 1. Unload the CNTL Dataset” on page 1-6 of this chapter, and Appendix A, “Distribution Tape Contents,” in this Guide.
2	Modify and execute INSTALL member.	✓		See “Step 2. Modify and Execute Install Member” on page 1-7 of this chapter.
3	APF authorize load library.	✓		See “Step 3. APF Authorize the Load Library” on page 1-7 of this chapter.
4	Bind Shadow Server product plans (only if you plan to use Shadow Server with DB2).	✓		See “Step 4. Bind Shadow Server Product Plans” on page 1-8 of this chapter.
5	Create VSAM datasets.	✓		See “Step 5. Create the VSAM Datasets” on page 1-9 of this chapter.
6	Set up started task JCL.	✓		See “Step 6. Set Up the Started Task JCL” on page 1-10 of this chapter.
7	Provide VTAM definitions.	✓		See “Step 7. Provide VTAM Definitions” on page 1-11. of this chapter.
8	Define TCP/IP port number.	✓		See “Step 8. Define TCP/IP Port Number (TCP/IP Only)” on page 1-13 of ths chapter.
9	Customize initialization EXEC.	✓		See “Step 9. Customize Initialization EXEC” on page 1-14 of this chapter..
10	Set up ISPF dialogs.	✓		See “Step 10. Set Up the ISPF/SDF Dialogs” on page 1-19 of this chapter.
11	Define started task name to security product.	✓		See “Step 11. Define the Started Task Name to Your Security Product” on page 1-20 of this chapter.
12	Create ODBC-optimized catalogs.		✓	See “Step 12. Create ODBC-Optimized Catalogs” on page 1-21 of this chapter.
13	Provide access to Shadow Server resources.		✓	See “Step 13. Provide Access to Shadow Server Resources” on page 1-22 of this chapter, and Appendix B, “Providing Access to Shadow Resources,” in this Guide.
14	Set up Shadow logging.		✓	See “Step 14. Set up Shadow Logging” on page 1-22 of this chapter, and Appendix C, “Shadow Logging,” in this Guide.

Table 1–1.

Shadow Server Installation				
No.	**Step**	**Req.**	**Opt.**	**Reference**
15	Enable SDF SMF recording.		✓	See “Step 15. Enable SMF Recording” on page 1-23 of this chapter.
16	Configure IMS connectivity options.		✓	See “Step 16. Configure IMS Connectivity Options” on page 1-23 of this chapter, and Appendix D, “IMS Connectivity Options (Includes AutoHTML),” in this Guide.
17	Install Transaction Server for CICS.		✓	See “Step 17. Install the Transaction Server for CICS” on page 1-23 of this chapter, and Appendix E, “Transaction Server for CICS,” in this Guide.
18	Install the Shadow_VSAM component for CICS.		✓	See “Step 18. Install Shadow_VSAM for CICS” on page 1-23 of this chapter, and Appendix N, “Shadow_VSAM for CICS,” in this Guide.
19	Install the Database Server for ADABAS.		✓	See “Step 19. Install the Database Server for ADABAS” on page 1-24 of this chapter.
20	Implement Shadow Virtual Connection Facility.		✓	See “Step 20. Implement the Shadow Virtual Connection Facility (VCF)” on page 1-24 of this chapter.
21	Configure load balancing.		✓	See “Step 21. Configure Load Balancing” on page 1-25 of this chapter, and Appendix H, “Load Balancing,” in this Guide.
22	Configure load balancing for Transaction Server for CICS.		✓	See “Step 22. Configure Load Balancing for Transaction Server for CICS” on page 1-26 of this chapter, and Appendix E, “Transaction Server for CICS,” in this Guide.
23	Configure EXCI failover.		✓	See “Step 23. Configure EXCI Failover” on page 1-26 of this chapter, and Appendix E, “Transaction Server for CICS,” in this Guide.
24	Configure Secure Sockets Layer (SSL).		✓	See “Step 24. Configure Secure Sockets Layer (SSL) Support” on page 1-26 of this chapter, and Appendix F, “Configuring Secure Sockets Layer (SSL) Support,” in this Guide.
25	**Not Applicable to Shadow Server**			
26	Issue the START command and verify initialization.	✓		See “Step 26. Start Shadow Server” on page 1-27 of this chapter.
27	Set up server to run under TSO		✓	See “Step 27. Set the Server to Run Under TSO” on page 1-27 of this chapter, and Appendix M, “Setting up Shadow Server to Run under User’s TSO Address Space,” in this Guide.

Table 1–1.

Shadow Server Installation

No.	Step	Req.	Opt.	Reference
28	Activate RRSAF and 2-phase commit support.		✓	See “Step 28. Activate RRSAF and Two Phase Commit Support” on page 1-28 of this chapter, and Appendix O, “Recoverable Resource Manager Services Attachment Facility (RRSAF) Support,” in this Guide.

Table 1–1.

Shadow OS/390 Web Server

These steps are all discussed in detail either in this chapter or in one of the appendices (see Reference column in Table 1–2 on page 1-4).

Shadow OS/390 Web Server Installation

No.	Step	Req.	Opt.	Reference
1	Unload the CNTL dataset.	✓		See “Step 1. Unload the CNTL Dataset” on page 1-6 of this chapter, and Appendix A, “Distribution Tape Contents,” in this Guide.
2	Modify and execute INSTALL member.	✓		See “Step 2. Modify and Execute Install Member” on page 1-7 of this chapter.
3	APF authorize load library.	✓		See “Step 3. APF Authorize the Load Library” on page 1-7 of this chapter.
4	Bind Shadow Server product plans (only if you plan to use Shadow Server with DB2).	✓		See “Step 4. Bind Shadow Server Product Plans” on page 1-8 of this chapter.
5	Create VSAM datasets.	✓		See “Step 5. Create the VSAM Datasets” on page 1-9 of this chapter.
6	Set up started task JCL.	✓		See “Step 6. Set Up the Started Task JCL” on page 1-10 of this chapter.
7	**Not Applicable to Shadow OS/390 Web Server**			
8	Define TCP/IP port number.	✓		See “Step 8. Define TCP/IP Port Number (TCP/IP Only)” on page 1-13 of ths chapter.
9	Customize initialization EXEC.	✓		See “Step 9. Customize Initialization EXEC” on page 1-14 of this chapter..
10	Set up ISPF dialogs.	✓		See “Step 10. Set Up the ISPF/SDF Dialogs” on page 1-19 of this chapter.

Table 1–2.

Shadow OS/390 Web Server Installation

No.	Step	Req.	Opt.	Reference
11	Define started task name to security product.	✓		See “Step 11. Define the Started Task Name to Your Security Product” on page 1-20 of this chapter.
12	**Not Applicable to Shadow OS/390 Web Server**			
13	Provide access to Shadow Server resources.		✓	See “Step 13. Provide Access to Shadow Server Resources” on page 1-22 of this chapter, and Appendix B, “Providing Access to Shadow Resources,” in this Guide.
14	**Not Applicable to Shadow OS/390 Web Server**			
15	**Not Applicable to Shadow OS/390 Web Server**			
16	Configure IMS connectivity options.	✓		See “Step 16. Configure IMS Connectivity Options” on page 1-23 of this chapter, and Appendix D, “IMS Connectivity Options (Includes AutoHTML),” in this Guide.
17	Install Transaction Server for CICS.		✓	See “Step 17. Install the Transaction Server for CICS” on page 1-23 of this chapter, and Appendix E, “Transaction Server for CICS,” in this Guide.
18	Install the Shadow_VSAM component for CICS.		✓	See “Step 18. Install Shadow_VSAM for CICS” on page 1-23 of this chapter, and Appendix N, “Shadow_VSAM for CICS,” in this Guide.
19	Install the Database Server for ADABAS.		✓	See “Step 19. Install the Database Server for ADABAS” on page 1-24 of this chapter.
20	Implement Shadow Virtual Connection Facility.		✓	See “Step 20. Implement the Shadow Virtual Connection Facility (VCF)” on page 1-24 of this chapter.
21	Configure load balancing.		✓	See “Step 21. Configure Load Balancing” on page 1-25 of this chapter, and Appendix H, “Load Balancing,” in this Guide.
22	Configure load balancing for Transaction Server for CICS.		✓	See “Step 22. Configure Load Balancing for Transaction Server for CICS” on page 1-26 of this chapter, and Appendix E, “Transaction Server for CICS,” in this Guide.
23	Configure EXCI failover.		✓	See “Step 23. Configure EXCI Failover” on page 1-26 of this chapter, and Appendix E, “Transaction Server for CICS,” in this Guide.
24	Configure Secure Sockets Layer (SSL).		✓	See “Step 24. Configure Secure Sockets Layer (SSL) Support” on page 1-26 of this chapter, and Appendix F, “Configuring Secure Sockets Layer (SSL) Support,” in this Guide.

Table 1–2.

Shadow OS/390 Web Server Installation

No.	Step	Req.	Opt.	Reference
25	Enable Web transactions (Auto HTML).		✓	See “Step 25. Web-enable IMS Transactions” on page 1-27 of this chapter, and Appendix P, “AutoHTML - Web Enabling Transactions (SWS),” in this Guide.
26	Issue the START command and verify initialization.	✓		See “Step 26. Start Shadow Server” on page 1-27 of this chapter..
27	Set up server to run under TSO		✓	See “Step 27. Set the Server to Run Under TSO” on page 1-27 of this chapter, and Appendix M, “Setting up Shadow Server to Run under User’s TSO Address Space,” in this Guide.
28	Activate RRSAF and 2-phase commit support.		✓	See “Step 28. Activate RRSAF and Two Phase Commit Support” on page 1-28 of this chapter, and Appendix O, “Recoverable Resource Manager Services Attachment Facility (RRSAF) Support,” in this Guide.

Table 1–2.

Installation

The following sections will guide you through the basic installation of the Shadow Server products. Information specific to Shadow Logging, Shadow Resources, IMS, CICS, VCF, SSL, DSN3@ATH exit, maintenance tapes, as well as the distribution tape contents can be found in the appendices at the end of this manual.

Note:

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

Step 1. Unload the CNTL Dataset

The Shadow Server distribution tape is available in standard IEBCOPY format and contains 31 libraries. See Appendix A, “Distribution Tape Contents,” for more information. The first library, the CNTL dataset, contains the JCL needed for the rest of the installation process. To unload it, use the following JCL (or equivalent):

```
//...          JOB
//UNLOAD       EXEC PGM=IEBCOPY
//TAPCNTL      DD   DSN=NEON.CNTL,DISP=(OLD,PASS),
//             UNIT=TAPE,VOL=SER=NSnnnn,
//             LABEL=(1,SL,EXPDT=98000)
//DSKCNTL      DD   DSN=prefix.NEON.CNTL,DISP=(NEW,CATLG),
//             UNIT=SYSDA,VOL=SER=??????,SPACE=(CYL,(1,1,25))
```

```
//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
//
```

Note:

The Volume Serial Number: The tape contains a serial number of the form “`NSnnnn`” on its external label. Use this label in the JCL above and in the INSTALL member in Step 2. Modify and Execute Install Member.

Step 2. Modify and Execute Install Member

Modify the INSTALL Member

Once you have unloaded the CNTL dataset, modify the `'INSTALL'` member as follows:

1. Change the job card for your data center’s standards.
2. Change the TAPVOL parameter to the volume serial written on the Shadow Server distribution tape.
3. If `TAPE` is not the correct unit name, change the TAPEUNT parameter.
4. Change the DISKPFX parameter to the high-level dataset qualifier you are using for Shadow Server libraries. The default is SDB.
5. If 3390 cannot be used to refer to the DASD unit on which Shadow Server will reside, change the DISKUNT parameter.
6. Change the DISKVOL parameter to the volser of the DASD volume on which the Shadow Server libraries will reside.

Submit the ’INSTALL’ Member

After completing the above modifications, submit `'INSTALL'` for execution. This member unloads the rest of the tape. Check the output of the IEBCOPY step carefully. Checking the condition code may not be sufficient, since IEBCOPY may return a condition code of zero even if nothing was copied.

Step 3. APF Authorize the Load Library

The Shadow 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.*

> ***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 MVS console. Use the following syntax with PROGxx:

```
APF ADD DSNAME(NEON.SV040100.LOAD) VOLUME(xxxxxx)
```

Step 4. Bind Shadow Server Product Plans

If you plan to use Shadow Server with DB2, you must use a DB2 application plan before you can use the SQL functions within the product; otherwise, you do not need to perform this step. Member `'BIND'` of the `'NEON.SV040100.CNTL'` dataset is used to bind the application plan to DB2.

If you choose not to bind either the Shadow Direct or Shadow OS/390 Web Server product plans, in Step BIDN0001 you can comment out the SYSTSIN line that points to member BINDW&DB2REL for Shadow OS/390 Web Server and BINDD&DB2REL for Shadow Direct. You will then need to update GRANT0001 and comment out the line that points to member GRANTW for Shadow OS/390 Web Server and GRANTD for Shadow Direct.

At the bottom of the JCL are certain fields which will need to be modified before submitting the JCL. The following parameters should be set:

SOUT	SYSOUT class for the output of the job
DISKUNT	Default unit name for DASD
HILEV	High level qualifier of the Neon datasets
DB2REL	DB2 release level
DSNLOAD	Name of the DB2 load library
DSNEXIT	Name of the DB2 exit library

* A site that does not want to IPL can use an existing authorized library, or it can use the MVS command or any one of the major on-line MVS performance/operations enhancement tools to add an entry for a new authorized library.

RUNLIB Name of the DB2 utility library, which must contain the DSNTIAD utility

Once the above parameters have been set, edit the `'DB2ID'` member and change the default DSN2 to the subsystem id of the DB2 in which the binds take place. This job should be run against all DB2 subsystems accessed by Shadow Server.

Next edit the `'DBTI??'` member, where `??` refers to the DB2 version you currently have installed. Make sure the plan name and program name are correct for the batch bind utility, DSNTIAD.

Finally, the BIND job automatically grants execute on the Shadow Server Product Plans to public. To change these grants, edit the `'GRANT'` member of the `'NEON.CNTL'` dataset. You can also issue these grants from DB2's SPUFI application.

The BIND job binds four Shadow Server product plans, SWSC1010, SWSR1010, SDBC1010, and SDBR1010. SWSC1010 and SDBC1010 are bound using Cursor Stability, while SWSR1010 and SDBR1010 are bound using Repeatable Read. It is recommended that you use SWSC1010 as the default Shadow OS/390 Web Server Product Plan and SDBC1010 as the default Shadow Direct Product Plan. Use SWSR1010 and SDBR1010 for operations that require Repeatable Read. The name of the plan **must** be SDxC1010, SWxC1010, SDxR1010 and SWxR1010.The x usually refers to the last character of the Shadow Server subsystem. The Bind job will use the default names as described above. If you want to change these plan names, edit the `'BIND??'` member, where `??` refers to the DB2 version. Change all plan names from their defaults to the new name.

Note:

If the 4th character of the Plan Name is an R, the NEON Client ODBC driver assumes that your application is using a plan where the plan was bound using an Isolation value of Repeatable Read. If you are not using Repeatable Read please ensure that your plan name does NOT have an R in the 4th character of the plan name as does the Shadow default plan SDBR1010. If it is any other character than an R we assume the plan was bound with an Isolation Level of Cursor Stability.

Step 5. Create the VSAM Datasets

Shadow Server keeps track of communication and SQL processing events and records this information in the trace dataset. You can view this information using the trace browse application. The Shadow Event Facility (SEF) supports Global variables that allow for storage of frequently used variables and keeps this information in the `'SYSCHKx'` datasets.

To define the trace dataset and the `'SYSCHKx'` datasets, edit and run the IDCAMS define command located in the `'DEFDIV'` member of the CNTL library. Modify the following items:

- Change `'NEON.SV040100.SDBB.TRACE'` to the name you will use for the trace dataset. This name needs to be defined to the `'SDBTRACE'` DD name in the started task JCL. If you are using SEF (required for Shadow OS/390 Web Server), change the two `'NEON.SV040100.SDBB.SYSCHKx'` names to the name you will use for global variable support. SDBB should be the same as the Shadow Direct subsystem name.
- `'NEON.SV040100.SWSS.TRACE'` and `'NEON.SV040100.SWSS.SYSCKx'` must also be changed to the same naming conventions as above for the Web Server component of Shadow. SWSS should be the same as the Shadow OS/390 Web Server subsystem name.
- The CYLINDER parameter for the trace dataset should contain space for the number of messages that you have specified in the BROWSEMAX parameter. You can count on 803 messages per 3390 cylinder and 669 messages per 3380 cylinder (each message taking up 918 bytes).
- The VOL parameter should specify the volume serial number on which the dataset will reside.
- The data component name (the last dataset name) should be the same as the cluster name, with an additional qualifier of “DATA”.

The easiest way to execute the **DEFINE CLUSTER** for the trace dataset and `'SYSCHK'` datasets 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.SV040100.CNTL(DEFDIV)'
```

Alternatively, you could include the **DEFINE CLUSTER** command in an IDCAMS step of a batch job.

Step 6. Set Up the Started Task JCL

The `'SDBB'` and `'SWSS'` members of the CNTL library contain the JCL procedure needed to run the Shadow Direct Server main address space (started task) and the Shadow OS/390 Web Server address spaces respectfully. The `'SDBB'` and SWSS PROC must be placed in a procedure library that will be searched for by the MVS **START** command (this may be `'SYS1.PROCLIB'`, but does not have to be).

Note:
The restriction in Shadow subsystem names is `'SDBx'` and `'SWSx'` for Shadow Direct and Shadow OS/390 Web Server, respectively, where `x` can be any ALPHABETIC character (no numerals). The subsystem name is specified via the `SSID` parameter in the started task JCL.

You must tailor the JCL (found in the ′SDBB′ and ′SWSS′ members) as follows:

1. Change the ′DB2LIB′ parameter to contain the name of the DB2 library, respectively, ensuring that the DB2 library is 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, update the ′IMSLIB′ parameter with the name of the IMS RESLIB dataset, and uncomment the parameter along with the STEPLIB definition.

3. Change the HLQ parameter to contain the high-level qualifier name of the Shadow Server libraries. This should properly set the Shadow 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, then use the ′NEON.EXECFB′ dataset. If they are RECFM=VB use the ′NEON.EXEC′ dataset. This dataset MUST contain the ′SDBxIN00′ and ′SWSxIN00′ initialization execs which will be modified in Step 9. Customize Initialization EXEC.

5. If you wish to run the sample VSAM RPC IVP, modify and execute the member ′DEFSTAFF′ in the ′NEON.SV040100.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 Server. For more information about the sample VSAM RPC, please refer to the *Shadow Server User's Guide*.

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

Step 7. Provide VTAM Definitions

If you are using SNA communications between Shadow Server systems, you will need to provide certain resource definitions for VTAM.

- APPL statements that define the Shadow 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 needs to be provided for Shadow 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=CONV,
                DSESLIM=20
                DMINWNL=10
                DMINWNR=10
                MODETAB=SDB2MODE
```

Where:

AUTH=(ACQ)

Permits Shadow Server to issue the OPNDST macro. This macro allows Shadow Server to acquire sessions with other Shadow Servers running on different systems.

applid Is the VTAM application name for the Shadow 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.

APPC=YES

Permits Shadow Server to use APPCCMD (LU 6.2) macros to communicate with other Shadow Server systems.

SECACPT=CONV

Allows certain security information to be accepted by Shadow 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 15. 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 the next section. This parameter must be supplied. 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.SV040100.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
```

The source for this sample can be found in the `'SDBMODE'` member of NEON.SV040100.CNTL.

Step 8. Define TCP/IP Port Number (TCP/IP Only)

If you are configuring Shadow Server to use TCP/IP, you must define a TCP/IP port number for Shadow Server to use 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, which essentially reserves this port number for the exclusive use of the indicated Shadow Server.

- The IBM TCP/IP port number definition is placed in the `'TCPIP.PROFILE'` dataset. This dataset is pointed to by the PROFILE dd name in the TCP/IP started task.

- The Interlink port number definition is placed in the `'INTCP.PARM(DNRSVC00)'` dataset. This dataset is pointed to by the SYSPARM dd name in the INTERLINK started task.

Step 9. Customize 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' for Shadow Direct and 'SWSxIN00' for Shadow OS/390 Web Server, where x is the last character of the 4 character subsystem ID.

Note:
The subsystem ID is usually 'SDBB' for Shadow Direct and 'SWSS' for Shadow OS/390 Web Server -- hence the EXEC is generally named 'SDBBIN00' or 'SWSSIN00'.

The EXEC must be placed in the library that is allocated to the 'SYSEXEC' DD name in the 'SDBB' and 'SWSS' Started Task procedure. We recommend that you place the 'SDBxIN00' and 'SWSxIN00' members in a separate dataset so that future maintenance will not accidentally overwrite your modified member.

The sample initialization EXEC that is shipped in member 'SDBBIN00' and 'SWSSIN00' of the 'NEON.EXEC(FB)' dataset is set up in such a way 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:

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

to:

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

and specify the APPLID for the value.

Sample EXECs

In the 'NEON.EXEC' REXX library, you will find the sample initialization EXECs, 'SDBBIN00' and 'SWSSIN00'. The 'SDBBIN00' and 'SWSSIN00' EXEC, when properly modified, will initialize your Shadow Server for client-server processing.

Initialization Exec Structure

You can make your initialization EXECs as simple or as complicated as you wish. However, there are a few general guidelines you should follow.

ADDRESS SDB and SWS Statement

Your initialization EXEC must include the following statements near the beginning of the program:

For Shadow Direct and 'SDBxIN00':

```
RC = SDRXIN()
ADDRESS SDB
```

For Shadow OS/390 Web Server and `'SWSxIN00'`:

```
RC = SWRXIN()
ADDRESS SWS
```

Note on Complex Logic

Your initialization EXEC can be customized to contain complex logic. For example, if you want to create just one EXEC for all your systems, you can use the REXX **SELECT** instruction to handle any local variations.

Note on Character Case

The initialization EXEC **must be completed in all upper case characters**. The only exception to this is with certain operand values, which can be coded in lower case if the actual operand is lower case. Do not code a lower case operand value if the actual value is upper case.

Specify Your License Code

You will need to specify your license code using the following line:

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

Here, you should replace the string "*LICENSECODESTRINGVAL*" with your personal license code. You can find this license code on the cover letter that you received with your copy of Shadow Server. 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.*

Define the Shadow ISPF Dialog Datasets

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

```
"MODIFY PARM NAME(EXECDSNAME)VALUE(NEON.SV040100.EXEC)"
"MODIFY PARM NAME(ISPLLIBDSNAME)VALUE(NEON.SV040100.LOAD)"
"MODIFY PARM NAME(ISPMLIBDSNAME)VALUE(NEON.SV040100.NEONMLIB)"
"MODIFY PARM NAME(ISPPLIBDSNAME)VALUE(NEON.SV040100.NEONPLIB)"
"MODIFY PARM NAME(ISPTLIBDSNAME)VALUE(NEON.SV040100.NEONTLIB)"
```

Even though the Shadow load library is allocated to Shadow 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

* Since the code is in encrypted form, you must be careful to enter the value exactly as given to you. Failure to do so will result in the inability to start the product on your machine. You can see the decrypted form of your license code after you have installed Shadow Server on your machine.

ISPF/SDF Dialogs). 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.

Define Shadow Event Facility (SEF) datasets

In order to use the Shadow Event Facility, the REXX datasets that contain the REXX rules need to be defined to Shadow Server. This is done by defining the prefix of the datasets followed by the suffix, leaving the distinguishing level as the wild card.

Note:
These definitions are not needed if you are not using SEF or the Shadow OS/390 Web Server.

```
“MODIFY PARM NAME(EPROPREFIX) VALUE(NEON.SV040100)”
“MODIFY PARM NAME(EPROSUFFIX) VALUE(EXEC)”
```

Define LU6.2 Connectivity (*Shadow Direct ONLY)*

In order for the Shadow client code to successfully communicate with Shadow Server via LU 6.2, the following **MODIFY PARM** command needs to be issued to define the VTAM applid.

The syntax is:

```
“MODIFY PARM NAME(APPLID) VALUE(APPLID)”
```

Where:

NAME Is 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.

VALUE Is the value of the parameter to be modified. In this case, the value must be the name you have defined to VTAM for the local SDB.

Note:
You must define to VTAM the VTAM application IDs used by Shadow Server before starting Shadow Server. VTAM applids are defined in members of the `’SDB.VTAMLST’` dataset. For more information, see “Step 7. Provide VTAM Definitions” on page 1-11.

Define TCP/IP Connectivity

If you are running MVS operating systems between MVS ESA SP5.2 and OS/390 2.4 with IBM’s TCP/IP stack 3.2, you have the option of using either OE TCP/IP or TCP/IP IUCV support. Customers running OS/390 2.5 and above with TCP/IP

3.4 or TCP/IP for OS/390 2.5 **MUST** use the IBM OE TCP/IP support since IUCV is no longer supported.

Define IBM TCP/IP IUCV Support

If you are using IBM’s TCP/IP IUCV support, you will need to define the TCP/IP started task name and port number to Shadow Server using the MODIFY PARM command.

Define the IBM TCP/IP started task name:

```
“MODIFY PARM NAME(TCPNAME) VALUE(address-space-name)”
```

Specify the port number that Shadow Server SSL will use. This is the same port number used in Step 7. Provide VTAM Definitions.

```
“MODIFY PARM NAME(IBMPORTNUMBER) VALUE(1200)”
```

Define SSL port number values:

```
  “MODIFY PARM NAME(IBMSSLPORTNUMBER)  VALUE(1300)”
```

Note:
SSL port number 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.

Define IBM OE Sockets Support

In order for Shadow to use OE TCP/IP, the Shadow Server started task id needs to be defined to OMVS:

RACF command:

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

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

If you are running OE sockets, you will need to assign port numbers to OE Sockets TCP/IP, using the **MODIFY PARM** command. OE Sockets can run over OE TCP/IP, MVS TCP/IP, and other TCP/IP implementations as well.

Define the OE and IBM port number values:

```
“MODIFY PARM NAME(TRACEOERW)      VALUE(YES)”
“MODIFY PARM NAME(OEPORTNUMBER)    VALUE(1200)”
“MODIFY PARM NAME(IBMPORTNUMBER)  VALUE(0000)”
```

▷ ***Note:***
"0000" must be used as the value for the IBM port number parameter.

As an option, you can define values for SSL port numbers if SSL sessions are being used.

Define SSL port number values:

```
“MODIFY PARM NAME(OESSLPORTNUMBER)  VALUE(1300)”
```

▷ ***Note:***
SSL port number 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.

If you are running multiple IBM TCP/IP OE stacks, and you want this Shadow 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 `XXXX` is one of the SUBFILESYSTYPE values specified in the BPXPRMxx PARMLIB member.

Define Interlink's TCP/IP Support

If you are using Interlink's TCP/IP, you must define the Interlink TCP/IP subsystem name and port number to Shadow Server.

First, define the Interlink subsystem name:

```
“MODIFY PARM NAME(ITCSUBSYSTEM) VALUE(ACSS)”
```

Define the Interlink port number:

```
“MODIFY PARM NAME(ITCPORTNUMBER) VALUE(1200)”
```

Define SSL port number values:

```
  “MODIFY PARM NAME(ITCSSLPORTNUMBER)  VALUE(1300)”
```

▷ ***Note:***
SSL port number 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.

Enabling IMS Support

Please refer to Appendix D, “Installing IMS Connectivity Options (Includes AutoHTML),” for information on how to enable IMS support.

Enabling CICS Support

Please refer to Appendix E, “Installing the Transaction Server for CICS,” for information on how to enable CICS support.

Step 10. Set Up the ISPF/SDF Dialogs

In order to invoke the ISPF/SDF application, use the following procedure:

1. For *S*hadow Direct, edit the `’SHADOW’` member of `’NEON.EXEC’`; for Shadow OS/390 Web Server edit the `’WEB’` member of `’NEON.EXEC’`, and change the parameter `llib` to read:
   `llib=<the Shadow load library>`

2. Copy the `’SHADOW’` and `’WEB’` EXECs 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 exec is invoked, as long as the Shadow Server is up and running.

When you invoke the SHADOW or WEB REXX execs for the first time, your TSO session will attempt to connect to the default Shadow subsystems, `’SDBB’` or `’SWSS’`. If you changed the subsystem name at install time, you need to invoke the `’SHADOW’` or `’WEB’` REXX execs with the SUB parameter in order to connect to the correct server. Failing to do this will cause the dynamic allocations of the Shadow ISPF datasets to fail and ISPF dialog errors to occur. Example:

```
SHADOW SUB(SWSA)  or WEB SUB(SWSA)
```

Once in the ISPF dialogs, you can use option 5.1 to switch between different Shadow subsystems (if you have installed more than one). Switching between Shadow OS/390 Web Server and Shadow Direct or vice-versa via option 5.1 is not supported.

Note:
If you are using TSO Command limiting, a feature of ACF/2 which requires access permissions to execute TSO commands, then you will need to define all the Shadow Server ISPF TSO commands to your security product before you can use the Shadow ISPF/PDF dialogs. Failure to do so will result in “`SDB or SWS command not found`” error messages when attempting to execute the ISPF/PDF dialogs. See Appendix B, “Providing Access to Shadow Resources,” for a list of TSO commands.

Step 11. Define the Started Task Name to Your Security Product

If you are running a security product (RACF, ACF/2, Top Secret), you may have to define userids for the Shadow Server address space and set up access rules so that Shadow Server can use the datasets it needs.

The table below summarizes the access requirements for Shadow Server as distributed.

Dataset Name	Access
IBM.DB2LIB	Execute
NEON.SV040100.LOAD	Execute
NEON.SV040100.EXEC	Read
NEON.SV040100.SDBB and SWSS.TRACE	Read, Write
NEON.SV040100.SDBB and SWSS.SYSCHK1 and SYSCHK2	Read, Write
NEON.SV040100.SDBB and SWSS.*.EXEC	Read, Write
NEON.SV040100.RPCLIB	Read, Execute
NEON.SV040100.SAMPLE.DATA (Web Server Only)	Read, Write
QUICKREF.LINKLIB*	Execute
QUICKREF.DATABASE*	Read

*. 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.

If you customize the dataset names for your installation, be sure to give those dataset names the appropriate access, and not the ones listed above.

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 Server. This is true whether or not the two Shadow Servers are on the same machine.

Note:

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

Define the started task to RACF

This section describes how to setup RACF to allow access to Shadow Server.

Define the name of the Shadow started task to RACF:

Example:

```
ADDUSER SDBB OWNER(PROD) DFLTGRP(PROD)
NAME('SHADOWDIRECT') DATA('xxxxxxxxxxxxxxx')

ADDUSER SWSS OWNER(PROD) DFLTGRP(PROD)
NAME('SHADOWWEBSERVER') DATA('xxxxxxxxxxxxxxx')
```

Define the started task to CA-Top Secret

In order to properly define *Shadow Direct* to CA-Top Secret, perform the following steps:

1. Set up the FACILITY entry for Shadow 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=xxx2IN,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)
```

   Where xxx is SDB for Shadow Direct and SWS for Shadow OS/390 Web Server.

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

   ```
   TSS ADDTO(SDBB) MASTFAC(SHADOWT)
   TSS ADDTO(SWSS) 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 12. Create ODBC-Optimized Catalogs

If you are using Shadow Server with Shadow Direct, you should create some ODBC-Optimized catalogs for your site. These catalogs will enhance the user-response time of your client desktop productivity tools. In some cases, these tables return information literally thousands of times faster than the standard IBM catalog tables.

Note:

These tables require some maintenance. Please refer to *Shadow Server User's Guide* for information on updating the ODBC-optimized catalogs.

The product distribution JCL library (`'NEON.SV040100.CNTL'`) has a member, `'BUILDTBL'`, that contains the SQL needed to create the ODBC-optimized catalogs. The SQL statements in this file are intended to be used with SPUFI; however, they can be altered for use with any other DB2 utility.

To build the optimized catalog, follow these steps:

1. If necessary, replace the string "SHADOW" with the authorization ID chosen by your site for the ODBC-optimized catalogs.
2. Replace the string "USERDB" with the name of an actual database.
3. Modify the grants as needed to provide access for the correct group of users.
4. Execute the SQL script.
5. Check the SPUFI output to see if any of the **INSERT** statements fail because no records satisfy the specified criteria. If this occurs, comment out these **INSERT** statements and re-execute the SQL script.

Note:

A separate set of optimized catalogs must be created for each DB2 system.

Step 13. Provide Access to Shadow Server Resources

Shadow Server provides protection for its resources using RACF classes and ACF2 generalized resource rules. See Appendix B, "Providing Access to Shadow Resources," in the *Shadow Installation Guide f*or more information.

Step 14. Set up Shadow Logging

Shadow Logging is a facility of Shadow Direct that allows capturing of MVS performance information into a set of DB2 tables and SMF. With Shadow Logging enabled, you can gather precise resource consumption information for your MVS Client Server environment.

Enabling Shadow Logging is required if you plan to use NEON's *Shadow Activity Monitor.* This product provides real-time data collection and monitoring of ODBC and Web Browser applications that access MVS with Shadow Direct or Shadow OS/390 Web Server through a point and click graphical interface.

See Appendix C, “Shadow Logging,” in the *Shadow Installation Guide* for more information.

Step 15. Enable SMF Recording

To enable SMF recording, set the following parameter within the Shadow Server initialization EXEC:

```
“MODIFY PARM NAME(SMFNUMBER) VALUE(xxxxxxxx)”
```

where `(xxxxxxxx)` is a number between 0 and 255. The default value for this parameter is 0, which indicates that no logging will take place. So, if you want to enable SMF recording, you will need to set the default value to the desired number.

You can also add or change the SMFNUMBER dynamically by using the ISPF panels (option 5.2, select the PRODTRACE parameter group and change the “SMF RECORD NUMBER” parameter).

For more information about this feature, see the “Supported SMF Fields” appendix in the *Shadow Server User’s Guide* and the *Shadow OS/390 Web Server User’s Guide*.

Step 16. Configure IMS Connectivity Options

This step is optional for Shadow Direct, but required for Shadow OS/390 Web Server.

The Transaction Server for IMS provides shrink-wrapped development tool access to IMS transactions without program changes. This step explains how to enables IMS support and configure the IMS Transaction Server to ODBC.

See Appendix D, “IMS Connectivity Options (Includes AutoHTML),” in the *Shadow Installation Guide* for more information.

Step 17. Install the Transaction Server for CICS

The Transaction Server for CICS provides shrink-wrapped development tool access to CICS transactions without program changes. See Appendix E, “Transaction Server for CICS,” in the *Shadow Installation Guide* for more information.

Step 18. Install Shadow_VSAM for CICS

The Shadow_VSAM component for CICS allows for the access and update of CICS assigned KSDS VSAM files. See Appendix N, “Shadow_VSAM for CICS,” in the *Shadow Installation Guide* for more information.

Step 19. Install the Database Server for ADABAS

The Database Server for ADABAS provides shrink-wrapped development tool access to ADABAS data without written host based programs.

To install the Database Server for ADABAS:

1. Edit the `'SDBxIN00'` (Shadow Direct) and `'SWSxIN00'` (Shadow OS/390 Web Server) execs in the EXEC library. Locate the following lines and change the IF statement as follows:

   ```
   /* ENABLE DATABASE SERVER FOR ADABAS */
   IF 1=1 THEN /* ENABLE ADABAS? */
   DO
      "MODIFY PARM NAME(ADABAS) VALUE(YES)"
   END
   ```

2. Edit your startup JCL for your Shadow Server by copying the ADALNK module from your `'ADABAS'` load library to one of the existing libraries on the STEPLIB ddname. If your load library is APF-authorized, you can add the ADABAS load library to the STEPLIB concatenation.

3. As an option, you can start your Database Server for ADABAS in read-only mode (i.e, only **SELECT**/**READ** operation is allowed). To do this, you need to add the following **READONLY** parameter to your `'SDBxIN00'` (Shadow Direct) and `'SWSxIN00'` (Shadow OS/390 Web Server) execs:

   ```
   "MODIFY PARM NAME(READONLY) VALUE(YES)"
   ```

Note:
With read-only mode, only **READ**/**SELECT** operation is allowed. You will get return code `-4025` if attempting **UPDATE**, **INSERT**, **DELETE**, **COMMIT**, **ROLLBACK** or **HOLD** operations on ADABAS files.

Step 20. Implement the Shadow Virtual Connection Facility (VCF)

VCF is a component of Shadow Direct which allows different types of connection modes to be used when connecting client applications to Shadow Server. With VCF, customers can implement applications that are scaleable to more efficiently handle large amounts of users, or large amounts of SQL statements.

VCF is implemented by having the VCF feature enabled in your license code, and by specifying one parameter in the IN00 file. This parameter is `HOSTFUNCTIONALLEVEL` and should be set to a minimum of 2, as shown below.

```
MODIFY PARM NAME(HOSTFUNCTIONALLEVEL) VALUE(2)
```

VCF is implemented on the client using a variety of parameters. These parameters are set in the datasource definition using the ADVANCED, MORE option. The parameters are listed below.

Connection Mode (CNMD)
Specifies the connection type to use. This parameter can be set to one of the following: PERMANENT(default), BLOCK, TRANSACTION, TRANSBLOCK and MESSAGE.

Message Type (MGTY)
Specifies the network protocol to use for messages. This parameter can be set to one of the following: NETWORK(default), MQ (not implemented yet), HTTP and HTTP/SSL.

Proxy Server Information (PXSR)
Specifies the name of the proxy server to be used if HTTP or HTTP/SSL is used as the messaging protocol.

Proxy Server Userid (PXUS)
Specifies the userid to logon to the proxy server when HTTP or HTTP/SSL is used as the messaging protocol.

Proxy Server Password (PXPW)
Specifies the password for the proxy server when HTTP or HTTP/SSL is used as the messaging protocol.

Step 21. Configure Load Balancing

To configure Load Balancing for Shadow Server:

1. Using the ISPF panels, check your product code string to make sure your license code includes load balancing:

   a. Start up your Shadow Server.
   b. Select option 5.2, Started Task Parameters, from the Shadow Primary Options menu.
   c. Select the PRODLICENSE group of parameters.
   d. Select the “PRODUCT FEATURE CODE STRING” parameter within this group. If there is no V in the product code string, then your Shadow Server has not been licensed for load balancing.

2. Add your Shadow Server to a group, by setting the following parameter within the Shadow Server initialization EXEC:

   ```
   “MODIFY PARM NAME(GROUPNAME) VALUE(xxxxxxxx)”
   ```

   You can also add or change a server’s group name dynamically by using the ISPF panels (option 5.2, select the PRODPARM parameter group and change the “LOAD BALANCING GROUP NAME” parameter).

   For more information about the Group Concept, see the “Load Balancing” appendix in the Shadow Server User’s Guide.

3. If you are using IBM TCP/IP 3.2 or earlier, you will need to set the following parameter within the Shadow Server initialization EXEC:

   ```
   "MODIFY PARM NAME(NETMODE) VALUE(TCPMAIN)"
   ```

For more information about Load Balancing, see Appendix I, “Load Balancing,” in your *Shadow Installation Guide.*

Step 22. Configure Load Balancing for Transaction Server for CICS

To enable CICS load balancing, you will need to set the following parameter within the Shadow Server initialization EXEC:

```
"MODIFY PARM NAME(CICSLOADBALANCE) VALUE(YES)"
```

You can also set this parameter dynamically by using the ISPF panels (option 5.2, select the PRODCICS parameter group and change the “USE TXN QDEPTH FOR LOAD BALANCING” parameter).

See Appendix E, “Transaction Server for CICS,” in your *Shadow Installation Guide* for more information.

Step 23. Configure EXCI Failover

If you want to add EXCI Failover, you will need to add the following parameter to the Define Connection statement in your IN00:

```
"ALTAPPLID(xxxxxxxx)"
```

where (xxxxxxxx) is the application id of your alternate CICS connection. The alternate CICS applid is used if your target CICS connection fails. The altapplid must have the same named connection definitions as well as application definition that are contained in the target CICS applid.

See Appendix E, “Transaction Server for CICS,” in your *Shadow Installation Guide* for more information.

Step 24. Configure Secure Sockets Layer (SSL) Support

To configure the Shadow Server for Secure Socket Layer (SSL) support, you should first complete installation of the server and verify that it can be started and stopped successfully.

See Appendix F, “Configuring Secure Sockets Layer (SSL) Support,” in the *Shadow Installation Guide* for more information.

Step 25. Web-enable IMS Transactions

See Appendix P, “AutoHTML - Web Enabling Transactions (SWS),” in the *Shadow Installation Guide* for information on how to accomplish this.

Step 26. Start Shadow Server

From an operational perspective, Shadow Server is an MVS started task. It can be started with the **START** command and stopped with the **STOP** command. In normal circumstances, Shadow 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 Server

To start Shadow Server, use the MVS **START** command:

```
S xxxx
```

where `xxxx` is `′SDBB′` for Shadow Direct or `′SWSS′` for Shadow OS/390 Web Server.

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

Shadow Server

To stop Shadow Server, use the MVS **STOP** command:

```
P xxxx
```

where xxxx is SDBB for Shadow Direct or SWSS for Shadow OS/390 Web Server.

Shadow Server will wait for all active conversations to end before terminating, so it may take a while to shut down. If you are in a hurry and can't wait for Shadow Server to terminate normally, use the **CANCEL** command:

```
CANCEL xxxx
```

where `xxxx` is `′SDBB′` for Shadow Direct or `′SWSS′` for Shadow OS/390 Web Server.

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

Step 27. Set the Server to Run Under TSO

See Appendix M, “Setting up Shadow Server to Run under User’s TSO Address Space,” in the *Shadow Installation Guide* for more information.

Step 28. Activate RRSAF and Two Phase Commit Support

Prerequisites

In order to run Recoverable Resource Services Attachment Facility (RRSAF) at your site, you will need the following:

- **DB2 at version 5.1 and later.** If a single copy of Shadow must also access a DB2 system at an earlier release, you cannot use RRSAF with that copy of Shadow.
- **Shadow Server v 4.5 or higher.** The server can use either CAF or RRSAF, but not both. You must choose the method at startup. It cannot be changed without restarting the server.
- **OS/390 RRS installed and running.** If RRS is not running, all RRSAF requests are rejected by DB2.
- **RRS requires System Logger Log Streams**, which can use the Coupling Facility or be DASD-only. To use DASD-only Log Streams, you must be at OS/390 release 2.4 or later, and put on the following two PTFs: UW43929 and UW43930. (Before OS/390 release 4, the System Logger did not support DASD-only Log Streams).

The best source for understanding what is needed to set up and run RRS is the IBM manual "*MVS Programming: Resource Recovery*" (GC28-1739), chapter 8 "Managing RRS". This chapter refers to the IBM manual "Setting Up a Sysplex", for instructions on how to setup and prepare System Logger Log Streams.

To activate RRS support:

Insert the following into the IN00 file:

```
/*---------------------------------------------------------------*/
/* RRS                                                           */
/*-+----1----+----2----+----3----+----4----+----5----+----6----*/
IF 1 = 1 THEN
  DO
   "MODIFY PARM NAME(DB2ATTACHFACILITY)   VALUE(RRS)"
   "MODIFY PARM NAME(PRODRRSGROUP)        VALUE(VISIBLE)"
   "MODIFY PARM NAME(RRS)                 VALUE(YES)"
   "MODIFY PARM NAME(RESOURCEMGRNAME)     VALUE(NEONRMAAAAAA)"
   "MODIFY PARM NAME(RECTABLEENTRIES)     VALUE(600)"
  END
```

Where:

DB2ATTACHFACILITY

Allows the user to control which mechanism to use for the db2 interface. The options are:

- The classic Call Attach Facility (CAF), using the DSNALI interface module.
- The new option of Recoverable Resource Services Attachment Facility (RRSAF), which can be used for DB2 v5.1 and above systems. The new facility allows the capability of a 2-phase commit through the attachment facility. Its interface routine is DSNRLI.

PRODRRSGROUP

Enables or disables the display of the PRODRRS parameter group.

RRS

Activates RRS support.

RESOURCEMGRNAME

Specifies the sysplex unique name of our RRS Resource Manger (which is an SDSRM). See the *IBM Programming: Resource Recovery* manual for valid naming conventions.

Note:

If the name is changed, any incomplete (in-doubt) transactions from the previous run will not be able to be completed.

RECTABLEENTRIES:

Specifies the number of entries the RRS recovery table must be able to hold. Entries in the RRS recovery table contain information about two-phase commit transactions that are in-doubt due to error conditions during transaction processing. the default is 400 entries, and the minimum value that will be used is 200 entries.

For more information about RRSAF, see Appendix O, “Recoverable Resource Manager Services Attachment Facility (RRSAF) Support,” in the *Shadow Installation Guide*.

Other Installation Issues

Setting up Two Shadow Servers of Different Versions or Maintenance Releases

See Appendix J, “Setting up Two Shadow Servers,” for information.

APPENDIX A:
Distribution Tape Contents

The following table lists the contents of the Shadow distribution tape. The Description column indicates whether files are specific to the Shadow Server or the Shadow OS/390 Web Server.

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 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
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 Direct	PO	FB	80	1	2
21	NEON.ATHEXECW	Web Server SEF ATH Library	PO	FB	80	1	8

No.	File Name	Description	DSORG	RECFM	LRECL	CYLS (3390)	DIR BLKS
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 Direct SEF Command Ruleset	PO	FB	80	1	4
33	NEON.CMDEXECW	Shadow Web Server SEF Command Ruleset	PO	FB	80	1	2
34	NEON.AHTML	Shadow Web Server Auto HTML	PO	VB	19036	2	2
35	NEON.CICSLOAD	CICS load library	PO	U	0	3	50

APPENDIX B: *Providing Access to Shadow Resources*

This appendix covers information about Shadow Server resources and providing resource protection.

Introduction

Shadow Server provides protection for its resources using RACF classes and ACF/2 generalized resource rules.

The overall class (or resource type, for ACF/2) for Shadow Server is specified with the RESOURCETYPE parameter found in the `'SDBxIN00'` initialization EXEC. If not explicitly specified, RESOURCETYPE defaults to `SDx`, where `x` is the fourth letter of the subsystem name (usually "`B`"). Because this first level qualifier is variable, it is possible to run multiple copies of Shadow Server, and just share the authorization rules, or keep them separate.

You may also use the RESOURCETYPE parameter to turn off product authorization checking. A value of "NON" disables all authorization checking.

For Shadow OS/390 Web Server the resource class ("resource type" for ACF/2) for URL authorization checking is specified with the URLRESOURCETYPE parameter found in the `'SWSxIN00'` initialization EXEC.

During initial installation of the product, NEON Systems recommends that you leave both of these parameter values set to "NON", **if possible**. This is because, during initial installation, most sites install the Shadow Server on test MVS systems, to which access is already limited and which is not directly exposed to the world-wide Internet. You may wish 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, the following security exposures may exist:

- Anyone with a valid TSO userid, may gain access to the Shadow Server ISPF control application and will be fully authorized to perform any function provided by the interface. This assumes, however, that the end-user has sufficient information at hand to logon to TSO/E and then gain access to the ISPF/SDF application.
- In the Shadow OS/390 Web Server anyone knowing the MVS host domain name and the TCP/IP port number will be able to run most of the NEON-supplied sample web applications from a web browser. The sample home page, HTML documentation files, and some simple, sample applications will be available to anyone with a TCP/IP link to your MVS host. (The sample

applications referred to here can be considered harmless in terms of accessing MVS resources.)

- In addition, anyone having a valid userid for your host MVS system (includes non-TSO users) will have sufficient authorization to access the Shadow OS/390 Web Server administration and control applications from a web browser, which perform some of the same functions as the ISPF-based interface.

Protected Resources

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

Resource Name	Description
CICSCONNECTIONS	Access to monitor and control CICS connections
CONTROLBLOCKS	Shadow Server internal data structures.
DATABASES	Databases that are defined to Shadow Server.
DATAMAP	Access to the Data Mapping Facility.
FILE	Shared Files which are defined to Shadow Server.
FILETYPE	Access to the server’s file-suffix/MIME-type control table.
IMSLTERM	Tables correlating to userid’s 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 Server.
PARMS	Access to the ISPF/SDF parm display. For Shadow OS/390 Web Server - used to control access to parameters accessible via the SWSPARM call, such as DEFAULTDB2SUBSYS and DEFAULDB2PLAN. If turning on Resource checking, give **READ** access to PARMS for all users using the SWSPARM call.
SDB (Shadow Direct Only)	Access to the ISPF/SDF interactive control facility.
SWS (Shadow OS/390 Web Server Only)	Access to the ISPF/SDF interactive control facility.
TOKENS (Shadow OS/390 Web Server Only)	Access to the ISPF Tokens Control Facility.
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

Resource Name	Description
TSO (Shadow OS/390 Web Server Only)	**Read/Execute** authorization is required to run a request procedure within an outbound TSO server address space or to look at the TSO control display panel. **Update/Write** authorization is required to terminate an Outboard TSO server address space using the TSO control display panel.
USERS	Access to the attached/remote users applications.
SEF	Access to the Shadow Event Facility dialogs.
TOKENS	Access to the Shadow Server Tokens Display.
RPC.<rpc_name>	RPC-based security (see “Providing Access to Shadow Resources,” on page B-6.)

How Resource Access is Determined

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

In order to expedite future authorization checks of an identical request, Shadow Server keeps the results of all security checks in protected storage; this expedites future authorization checking of any identical request.

Note:

The “look-aside” security check information is saved on a Task Control Block (TCB) basis, and remains in effect until the TCB terminates. Thus, if you attempt to access the Link Control application and are denied access, but then your security administrator grants you access, you will need to exit the ISPF/SDF application to terminate its TCB.

Depending on your security package, you may have to take other actions. Under ACF/2, for example, you will have to issue the **ACFRESET** command.

All security authorization events are logged in the trace browse facility. Furthermore, in cases where access is denied, a message is produced.

The type of access you are requesting (**ADD**/**ALTER**[*], **READ**, or **UPDATE**) depends upon which facility you are using. The table below shows the type of access that is required to use product facilities.

In the following tables, xxx refers to SDB for Shadow Direct and SWS for Shadow OS/390 Web Server.

[*] The ACF/2 ADD is equivalent to the RACF ALTER.

Shadow Server Facility	Suggested User	Resources	Access Required
Viewing product control blocks using ISPF/SDF option 5.3.	DBA, Program Products	CONTROLBLOCK, *xxx*	READ
Modifying product control blocks using a future facility.	DBA, Program Products	CONTROLBLOCK, *xxx*	UPDATE
Using the xxx command.	DBA, Program Products, VTAM, Operations	CONTROLBLOCK	READ
Defining Links using the **ADDRESS SDB DEFINE LINK** command.	DBA, Program Products, VTAM, Operations	xxx	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, *xxx*	UPDATE
Defining Databases using the **ADDRESS SDB DEFINE DATABASE** command.	DBA, Program Products	LINKS, *xxx*	ADD/ALTER
Viewing Databases using either ISPF/SDF option 2 or the ADDRESS xxx DISPLAY DATABASE command.	DBA, Program Products, VTAM, Operations	DATABASES	READ
Modifying Databases using either ISPF/SDF option 2 or the **ADDRESS xxx MODIFY DATABASE** command.	DBA, Program Products	DATABASES, *xxx*	UPDATE
Viewing Attached Users using either ISPF/SDF option 3 or the **ADDRESS xxx DISPLAY ATTACHED** command.	DBA, Program Products, VTAM, Operations	DATABASES, *xxx*	READ
Viewing Remote Users using either ISPF/SDF option 4 or the **ADDRESS xxx DISPLAY REMOTE** command.	DBA, Program Products, VTAM, Operations	USERS, *xxx*	READ
Killing Remote Users from the ISPF/SDF option 4 display.	DBA, Operations, Developers, End-Users	USERS, *xxx*	READ, UPDATE
Viewing product started task parameters using either ISPF/SDF option 5.2 or the **ADDRESS xxx DISPLAY PARM** command.	DBA, Program Products, VTAM, Operations	USERS, *xxx*	READ
Modifying product started task parameters using either ISPF/SDF option 5.2 or the **ADDRESS xxx MODIFY PARM** command.	DBA, Program Products, VTAM, Operations	PARMS, *xxx*	UPDATE
Viewing all trace browse data. See “Providing Access to Shadow Resources,” on page B-10.	DBA, Program Products, VTAM, Operations	PARMS, *xxx*	READ

Shadow Server Facility	Suggested User	Resources	Access Required
Issuing SQL statements via Shadow SPUFI	DBA, Program Products, VTAM, Operations	TRACEBROWSE, TRACEDATA, *xxx*	READ
Correlating userid’s or TCP/IP addresses to LTERMs	DBA, Shadow Administrator	IMSLTERM xxx	READ, UPDATE

Define Shadow Resources to RACF

The following steps describe how to define classes and resources to RACF.*

1. Define a new RACF class to the RACF Class Descriptor Table for Rxxy, where xx is equal to the first two characters of the Shadow subsystem name and y is equal to the last character of the Shadow subsystem name.

> ***Note:***
> Because RACF requires the class name to be a minimum of four 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 *
Rxxy     ICHERCDE CLASS=Rxxy,
               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)
/*
```

* For information on how to configure classes and resources for either ACF/2 or Top Secret, please consult their respective system programmer’s guides...

2. So that RACF will recognize the new class, an IPL is required to change the RACF Class Descriptor Table.

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

   ```
   SETROPTS CLASSACT(Rxxy)
   ```

4. Define all RACF resource types to class RSDx:

   ```
   RDEFINE Rxxy CONTROLBLOCKS UACC(NONE)
   ```

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

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

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

   where `AI38AAS` is the id of the user to whom you wish to grant **READ** permissions.

   Note:
   Repeat this **RDEFINE** 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.

*Define Security for RPCs (*Shadow Direct *Only)*

Shadow Direct supports RPC level security. If you need to restrict access to RPCs, please use the following procedure:

1. In the `'SDBxIN00'` initialization EXEC, add the following **MODIFY PARM** parameters:

   a. Turn on RPC security.

      ```
      "MODIFY PARM NAME(CHECKRPCAUTHORITY) VALUE(YES)"
      ```

   b. Turn on security resource checking.

      ```
      "MODIFY PARM NAME(RESOURCETYPE) VALUE(SDx)"
      ```

   c. **(Optional)** Turn on authorization event tracing (this allows you to trace the RPC security checks).

      ```
      "MODIFY PARM NAME(TRACEAUTHEVENTS) VALUE(YES)"
      ```

d. ACF/2 Only: In order to use RPC security with ACF/2, you need to be running with a version of ACF/2 that supports SAF calls. The following parameter enables Shadow Server to use SAF calls for Resource Rules.

```
“MODIFY PARM NAME(ACF/2SAFCALL) VALUE(YES)”
```

2. Define all RPCs to your security product.

```
RACF example) RDEFINE RSDx RPC.<rpc_name>UACC(NONE)
```

3. Grant access to the RPCs to the desired users.

```
(RACF example) PERMIT RPC.<rpc_name> CLASS(RSDx) ID(<tso_id) ACCESS(READ)
```

Note:
If you turn on ‘`Check RPC Authority`’ you will need to define each RPC to your security product with the steps above or the execution of the RPC will not be authorized.

Define Shadow Resources to CA-Top Secret

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

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

where `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) xxy(CONTROLBLOCKS)
```

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

3. Permit the resources to profiles or users:

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

where `xx` is the first two characters of the Shadow Server subsystem name and `y` is the last character of the Shadow Server subsystem name.

Define Shadow Resources to ACF/2

1. Define a generalized resource class name, `xxy`. Where `xx` is the first two characters of the Shadow Server subsystem name and *y* is the last character of the Shadow Server subsystem name. Example: SDB for Shadow Direct or SWS for Shadow OS/390 Web Server.

2. Define resource rules for each of the resource classes that Shadow supports. Member `'ACF/2DFN'` of the `'NEON.CNTL'` dataset can be used as an example.

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

3. Use the following ACF/2 command to allow users access to the resource rule:

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

Note:

Shadow OS/390 Web Server uses the MVS SAF interface via RACROUTE to perform generalized resource rule checking. In order to make the new resource types recognizable via the SAF interface, your security administrator must also issue the following commands under TSO:

```
TSO ACF
SET CONTROL(GSO)
INSERT CLASMAP.sws RESOURCE(sws) RSRCTYPE(sws)
INSERT CLASMAP.url RESOURCE(url) RSRCTYPE(url)
```

Define Shadow ISPF Load Modules

If you are using TSO command Limiting to restrict execution access to TSO commands, the following Shadow Direct and Shadow OS/390 Web Server ISPF load modules need to be defined to your security product:

Shadow Direct Load Modules:

SDADDM
SDADEX
SDB
SDBI
SDBICOMP
SDBIDB
SDBIMEX
SDBOB
SDBOCP
SDBORU
SDBTIMD
SDBVBFB
SDBX
SDBXCOMP
SDBXDB
SDBXSCAN
SDB2AUEX
SDB2IN
SDB2RU
SDDGRU
SDDGSP
SDHOCM
SDIMFU
SDISCBRU
SDISSTRU
SDISTBRU
SDLEPGLI
SDLESVRU
SDLINK
SDNTLDMD
SDRXBR
SDRXID
SDRXIN
SDRXLELK
SDRXPC
SDRXSG
SDRXSQ
SDRXST
SDRXTK
SDRXVA
SDSLSVMD
SDSLUTCC
SDSLUTCK
SDSLUTDE
SDSLUTKY
SDSLUTPA
SDSLUTRQ

Shadow OS/390 Web Server Load Modules:

SWS
SWSI
SWSICOMP
SWSIDB
SWSIMEX
SWSINFO
SWSOB
SWSTIMD
SWSX
SWSXCOMP
SWSXDB
SWSXSCAN
SWS2AUEX
SWS2IN
SWS2RU
SWDGRU
SWDGSP
SWHOCM
SWIMFU
SWISCBRU
SWISSTRU
SWISTBRU
SWLEPGLI
SWLESVRU
SWLINK
SWNTLDMD
SWRXBR
SWRXID
SWRXIN
SWRXLELK
SWRXPC
SWRXSG
SWRXSQ
SWRXST
SWRXTK
SWRXVA

Note on Started Task Security

A major exception to the security authorization scheme is the Shadow Server Started Task itself. All work performed under the product address space, on behalf of itself, is exempt from security. As a practical matter, this means that the `'SDBB'` or `'SWSS'` 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 Server trace log, SQL source statements* and binary data that underlies the trace log (i.e., control blocks). Users who have **READ** authority for the TRACEDATA resource (as well as **READ** authority for SDB/SWS 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 what kind event it is:

- **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 not 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.
- **Sensitive Data Censorship (*SWS Only*):** Potentially sensitive data, such a HTML form input variables, userids, passwords, and web transaction responses can be censored in the wrap-around trace. User's without TRACEDATA **READ** authorization will not be able to view the actual data contents. The degree to which censorship is applied can be selected depending on the needs of your installation. The CENSORURLQUERYDATA, CENSORURLAUTHDATA, CENSORHTTPRESP, CENSORAPIDATAVALUES start-up parameters allow for fine tuning of the censorship level.

*The SQL statements referred to here are the real SQL source statements (as taken from DBRM's or prepared strings) which may contain table name, column names, etc.

Resource Security for Test Version of Shadow Server

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

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

APPENDIX C: Shadow Logging

This appendix covers Shadow Logging, a facility of Shadow Direct.

Introduction

Shadow Logging allows capturing of MVS performance information into a set of DB2 tables and SMF. With Shadow Logging enabled, you can gather precise resource consumption information for your MVS Client Server environment. Please note that when enabling Shadow Logging, information is not only written to a set of DB2 tables, but additional Shadow type 01 and new type 02 records will be written out to SMF. Type 01 records are shared with our normal end-of-sessions records. These records can be distinguished via the SMO1RCTY field in the SMF type 01 record (see the *User's Guide* for more detailed information about SMF types).

With Shadow Logging, Shadow Server will, at a specified interval, write out total MVS resource usage information into a DB2 Intervals table for a specified time interval. It will also write out detailed information for each connection into a DB2 Sessions table. At client disconnect time, a record will also be written to a DB2 sessions table similar to the record already cut by Shadow Direct to SMF. With these features in place, detailed reporting can be accomplished for CPU resource consumption in your Client Server applications.

Shadow Logging also enables the use of the Shadow Activity Monitor, a GUI Client Server Performance and Reporting Monitor. Shadow Activity Monitor is included on the Neon Systems Installation CD. Please refer to the Shadow Activity Monitor documentation for further information.

Note:

In order to use the Shadow Logging feature a M feature code string is required in your Shadow License code. Feature Code strings can be viewed by accessing the Shadow ISPF/SDF panels, selecting option 5.2 for Shadow Server Main Task Parm Groups and then selecting PRODLICENSE. Your feature codes can be viewed by looking at the PRODUCT FEATURE CODE STRING parameter. Without a M in your license code only the SQL Logging feature can be enabled.

Shadow Logging Tables

The Shadow Logging tables are created by either running the script `‘BUILDLOG’`, which only builds logging tables or `‘BUILDLOG’`, which is part of the ODBC Optimized catalogs install. If you decide you do not want to use the ODBC optimized catalogs and only want to use the Shadow Logging feature, run only the creates for the Shadow Logging Tables, being careful to also create all the indexes and to run the grants necessary to allow the inserts and deletes to be carried out by Shadow Server (see LOGUSERID under Enabling Shadow Logging). The following tables are part of Shadow Logging:

Shadow.Intervals

This table contains precise MVS resource usage information for all connections that were active for an interval. Each record in the table is associated with a starting interval time. The information in each record is the total MVS resources used by ALL connections for the starting interval time until the next starting interval time.

Shadow.Sessions

This table contains precise MVS resource usage information for each connection. At the specified interval time, a record will be written to the Sessions table for every connection, detailing the MVS resource usage for each connection. Additionally, a Sessions record will be cut at client disconnect time detailing all the recordable CPU resources used for the entire connection.

Shadow.Sqlsource

This table contains all the SQL executed by Shadow Server. This feature can be enabled to capture SQL for information purposes, but its main purpose is for the future enhancements of DSA, the Dynamic to Static Analyzer.

Shadow.Urls

This table contains all URLs used to access Shadow OS/390 Web Server. This feature is currently not in use by Shadow Direct.

Shadow.Intervals

This table contains one record for each interval. Each record contains the following information:

RECORD_TYPE:	Describes the record type. Currently this will always be Summary.
TOTAL_CPUTIME:	Total CPUTime used by all connections.
DATABASE_CPUTIME:	Total Database_CPUTime used by all connections. Currently this is IMS and DB2. This field will not reflect CPU usage by RPCs that access IMS and DB2.

NETWORK_CPUTIME:	Total Network CPUTime trappable within Shadow Server. **Note:** Some Network CPUTime will not be trappable within Shadow Server. Therefore, this field may not reflect the total CPUTime.
REXX_CPUTIME:	Total CPUTime used by running SEF Rexx programs.
RPC_CPUTIME:	Total CPUTime used by RPCs. This includes CPU time used in DB2 and IMS if the RPCs accessed these databases.
OTHER_CPUTIME:	This is all unaccountable CPUTime and therefore is associated with Shadow itself.
USER_COUNT:	The number of users that were connected during this interval.
SMFID:	The Shadow SMFID as defined within Shadow Server.
PRODUCT_SUBSYSTEM:	The 4 character Shadow subsystem id.
INTERVAL_START:	The Interval Start time.
CONNECT_TIME:	N/A and is set to NULL.
BYTES_READ:	Total number of bytes sent up from the client connections.
BYTES_WRITTEN:	Total number of bytes of data written down to the client workstations.
COMMIT_COUNT:	Total number of commits performed.
ROLLBACK_COUNT:	Total number of Rollbacks performed.
SQL_COUNT:	Total number of SQL queries executed.
RPC_COUNT:	Total number of RPC executed.

Shadow.Sessions

This table contains one Interval record for each user for each interval. It also contains session records that contain total information for the entire connection. Session records get cut at client disconnect time, similar to when an SMF record is written. Each record contains the following information. CPUTimes, depending on the record type, will be either for the entire Session or for the Interval.

USERID:	USERID associated with the record
CLIENT_SYSTEM:	Client PC name
PROTOCOL:	Either TCP/IP or LU 6.2
RECORD_TYPE:	Either Session or Interval
TOTAL_CPUTIME:	Total CPUTime
DATABASE_CPUTIME:	Total Database_CPUTime used. Currently this is IMS and DB2. This field will not reflect CPU usage by RPCs that access IMS and DB2.

NETWORK_CPUTIME:	Total Network CPUTime trappable within Shadow Server. **Note:** Some Network CPUTime will not be trappable within Shadow Server, therefore this field may not reflect the total CPUTime.
REXX_CPUTIME:	Total CPUTime used by running SEF Rexx programs.
RPC_CPUTIME:	Total CPUTime used by RPCs. This time will include CPU time used in DB2 and IMS if the RPCs accessed these databases.
OTHER_CPUTIME:	This is all unaccountable CPUTime and therefore is associated with Shadow itself.
SMFID:	The Shadow SMFID as defined within Shadow Server.
PRODUCT_SUBSYSTEM:	The 4 character Shadow subsystem id.
DRIVER_VERSION:	The *NEON ODBC driver version.*
DRIVER_DATE:	The *NEON ODBC driver date.*
CONNECTION_ID:	The unique *Shadow Server connection id*
LOGON_TIME:	Time the user logged on.
LOGOFF_TIME:	Time the user logged off. If an Interval record, this value is NULL.
INTERVAL_START:	Interval Start time. If a Sessions record, this value is NULL.
CONNECT_TIME:	Number of seconds user was connected. If an Interval record, this value is NULL.
BYTES_READ:	Total number of bytes of data read from the client workstation.
BYTES_WRITTEN:	Total number of bytes written to the client workstation.
COMMIT_COUNT:	Total number of commits performed.
ROLLBACK_COUNT:	Total number of rollbacks performed.
SQL_COUNT:	Total number of SQL queried executed.
RPC_COUNT:	Total number of RPC executed.
ABEND_CODE:	Abend for the session if one occurred. For Interval records this value is NULL
IP_ADDRESS:	If a TCP/IP connection, this is the IP address of the client workstation, otherwise NULL.
LU_NAME:	If an LU 6.2 connection, the LU name used for the connection.
ORIGINAL_USERID:	Original USERID, recorded in case it was changed by a SEF rule.
PLAN:	DB2 plan used.
DATABASE:	DB2 subsystem connected to.
APPLICATION:	Application name, this value is set by the client’s ODBC connection information.
USERPARM:	Optional userparm from the client. This value is set by the client’s ODBC connection information.

Enable Shadow Logging

To enable Shadow Logging in Shadow Direct, set the following parameters in the Shadow Initialization EXEC:

```
“MODIFY PARM NAME(LOGDB2SUBSYS)               VALUE(DSN)”
“MODIFY PARM NAME(LOGINTERVALS)               VALUE(YES)”
“MODIFY PARM NAME(LOGSESSIONS)                VALUE(YES)”
“MODIFY PARM NAME(LOGSQLSOURCE)               VALUE(NO)”
“MODIFY PARM NAME(LOGURLS)                    VALUE(NO)”
“MODIFY PARM NAME(LOGUSERID)                  VALUE(SDBB)”
“MODIFY PARM NAME(LOGRETAININTERVALS)         VALUE(30)”
“MODIFY PARM NAME(LOGRETAINSESSIONS)          VALUE(30)”
“MODIFY PARM NAME(LOGRETAINSQL)               VALUE(30)”
“MODIFY PARM NAME(LOGRETAINURLS)              VALUE(30)”
“MODIFY PARM NAME(RECORDINGINTERVAL)          VALUE(900)”
```

LOGDB2SUBSYS

Needs to be set to the DB2 subsystem name that contains the Shadow Logging tables. It is highly recommended that you use only one DB2 subsystem for all of your Shadow Logging.

Note:

If your DB2 subsystem is down, Shadow Server will cache all failing inserts and retry every 5 minutes until successful. Therefore, no Logging data will be lost if your DB2 subsystem containing the DB2 logging tables is down.

LOGINTERVALS

Specifies whether to turn on or off the Intervals recording.

LOGSESSIONS

Specifies whether to turn on or off the Sessions recording.

LOGSQLSOURCE

Specifies whether to turn on or off the SQL recording.

LOGURLS Specifies whether to turn on or off the URL recording.

LOGUSERID

Specifies what USERID should be used to do the inserts into the logging tables. This USERID must have insert and delete authority for each of the Logging tables.

LOGRETAININTERVALS

Specifies how many days to retain information in the Intervals table. Each hour Shadow Direct runs a task to delete all records exceeding this value.

LOGRETAINSESSIONS

Specifies how many days to retain information in the Sessions table. Each hour Shadow Direct runs a task to delete all records exceeding this value.

LOGRETAINURLS

Specifies how many days to retain information in the URLS table. Each hour Shadow Direct runs a task to delete all records exceeding this value.

RECORDINGINTERVAL

Controls how often records are created. The interval value is specified in seconds and should be a factor of one hour. In other words, the value should divide evenly into 3600.

APPENDIX D: IMS Connectivity Options (Includes AutoHTML)

This appendix explains how to install and configure IMS support for Shadow Direct and Shadow OS/390 Web Server. IMS support includes:

- **IMS CCTL/DBCTL**
- **Transaction Server for IMS**

In addition, it provides information regarding Shadow_IMS, a generic RPC that allows you to invoke existing transactions.

For IMS programming information, see the *Shadow Programming Guide*.

Installation Prerequisite

Before proceeding to the installation procedures for IMS CCTLL/DBCTL and IMS Transaction Server, you will need to do the following:

1. Check to see if your Shadow license code includes IMS Support. To do this:

    a. Start up the Shadow Server, and go to the Primary Options menu.

    b. Select option 5.2, Started Task Parameters.

    c. Select the PRODLICENSE group of parameters.

    d. Select the "PRODUCT FEATURE CODE STRING" parameter within this group. If there is no I in the product code string, then your Shadow Server has not been licensed for IMS support.

2. Make sure the IMS recons are initialized at SHARECTL.

IMS CCTL/DBCTL Support

This section applies to both Shadow Server and Shadow OS/390 Web Server and consists of the following steps:

- Step 1. Modify Started Task.
- Step 2. Modify Installation EXEC.
- Step 3. Verify CCTL/DBCTL Installation.

Step 1. Modify Started Task

Before you begin, add the IMS `‘RESLIB’` to the RPC DD statement (`‘SDBRPCLB’` or `‘SWSRPCLB’`) if it is not already in the LPA or link list.

Step 2. Modify Installation EXEC

The following example demonstrates how to enable IMS support using the **MODIFY PARM** command in the `IN00`:

```
“MODIFY PARM NAME(DBCTL)            VALUE(YES)”
“MODIFY PARM NAME(PROCESS)          VALUE(11)”
“MODIFY PARM NAME(IMSMINTHREADS)    VALUE(5)”
“MODIFY PARM NAME(IMSMAXTHREADS)    VALUE(10)”
“MODIFY PARM NAME(IMSID)            VALUE(IVP1)” imsid of the
                                    DBCTL region
“MODIFY PARM NAME(IMSDSNAME)        VALUE(IMS.RESLIB)” name of
                                    the IMS RESLIB dataset
```

Where:

DBCTL

is used to initialize DBCTL support.

PROCESS

is the initial process block count. This parameter needs to be equal to IMSMAXTHREADS plus the number of users that will be using the Shadow ISPF/SDF dialogs.

IMSMINTHREADS

is the number of DBT threads to open initially when Shadow Connects to IMS

IMSMAXTHREADS

is the maximum number of allowed DTB threads to be active at one time.

IMSID

is the IMS identification of the DBCTL region.

IMSDSNAME

is the the DSN name of the DRA RESLIB.

Note:

Be sure to substitute your IMSID for *`IVP1`* and your IMSDSNAME for the *`IMS.RESLIB`*.

In addition to being changed by the `“MODIFY PARM”` command, some of these parameters can also be changed dynamically by using **option 5.2** and selecting

PRODIMS. For detailed information about these parameters, see the “Started Task Parameters” section located under the PARAMETERS tab of this binder.

Note:
The DBCTL and IMSID parameters are not updateable.

Step 3. Verify the IMS CCTL/DBCTL Installation

You will need to verify the IMS CCTL/DBCTL installation for both Shadow Server and Shadow OS/390 Web Server:

Shadow Direct

To verify that Shadow Server has successfully connected to IMS via CCTL, submit a IMS/RPC request from NEON Client:

1. Check the syslog of the Shadow control region for the following message:

   ```
   SDB4353I IMS CCTL SUPPORT ACTIVATED
   ```

   Note:
   If you do not see the above message on the syslog after the Shadow Server has started and initialized, do not proceed with the next steps.

2. To run the RPC demo program, `‘SDCOIM’`, you must compile the COBOL member, located in the `‘NEON.SAMP’` dataset. It is preferable to use LE370 COBOL. If this is not available, use COBOL II. The PART transaction delivered by IBM with IMS is used by the demo program and must be installed in your IMS system for this IVP to run.

3. Configure an ODBC data source. For further information, see the Shadow Server User’s Guide chapter, “Configuring Data Sources”.

4. Do one of the following, depending on the operating system of the client PC:

   – **Windows 95/NT.** From the client PC Start menu, select **Start**>**Programs**>**Shadow Direct**>**VB Demo Application**. The VB Demo window opens.

   – **Windows 3.x.** Select the Shadow group icon, then double-click the VB Demo application icon. The VB Demo window opens.

5. In the VB Demo window, select **Connections**>**Add New Connection**.

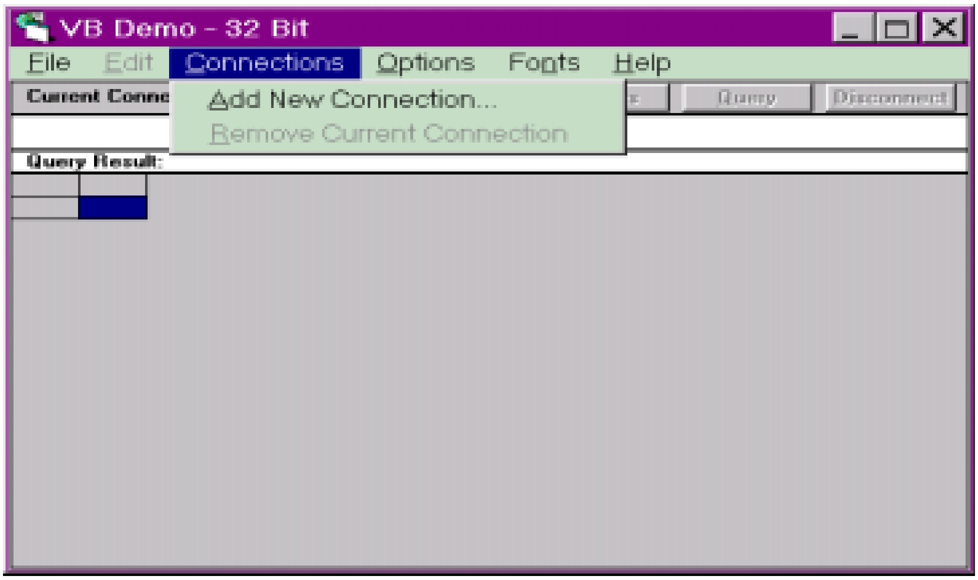
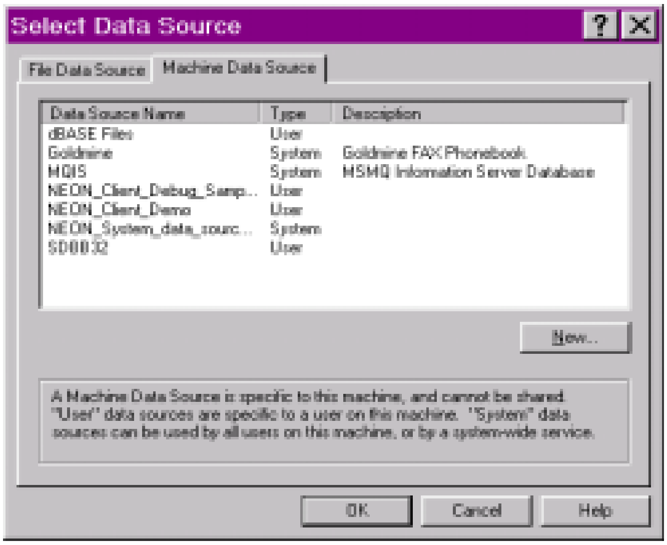


Figure D–1. VB Demo Screen -- Connections

The Select Data Source window opens.

6. Select a data source using either the File Data Source or Machine Data Source tab:



Data Source Name	Type	Description
dBASE Files	User	
Goldmine	System	Goldmine FAX Phonebook
MQIS	System	MSMQ Information Server Database
NEON_Client_Debug_Samp...	User	
NEON_Client_Demo	User	
NEON_System_data_sourc...	System	
SDBB32	User	

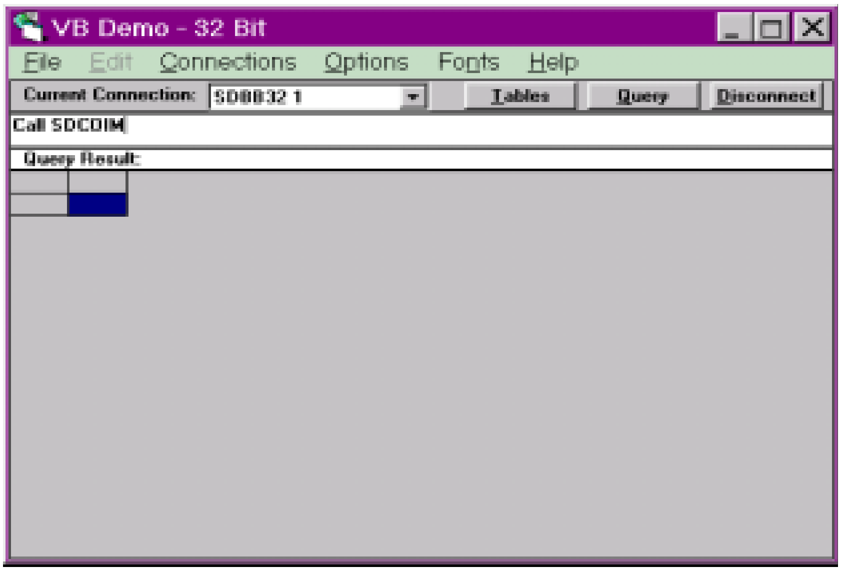
A Machine Data Source is specific to this machine, and cannot be shared. "User" data sources are specific to a user on this machine. "System" data sources can be used by all users on this machine, or by a system-wide service.

Figure D–2. Select Data Source Screen

Click OK. This will take you back to the main VB demo screen.

7. In the input box (the white area directly below the Current Connection listing), type the following:

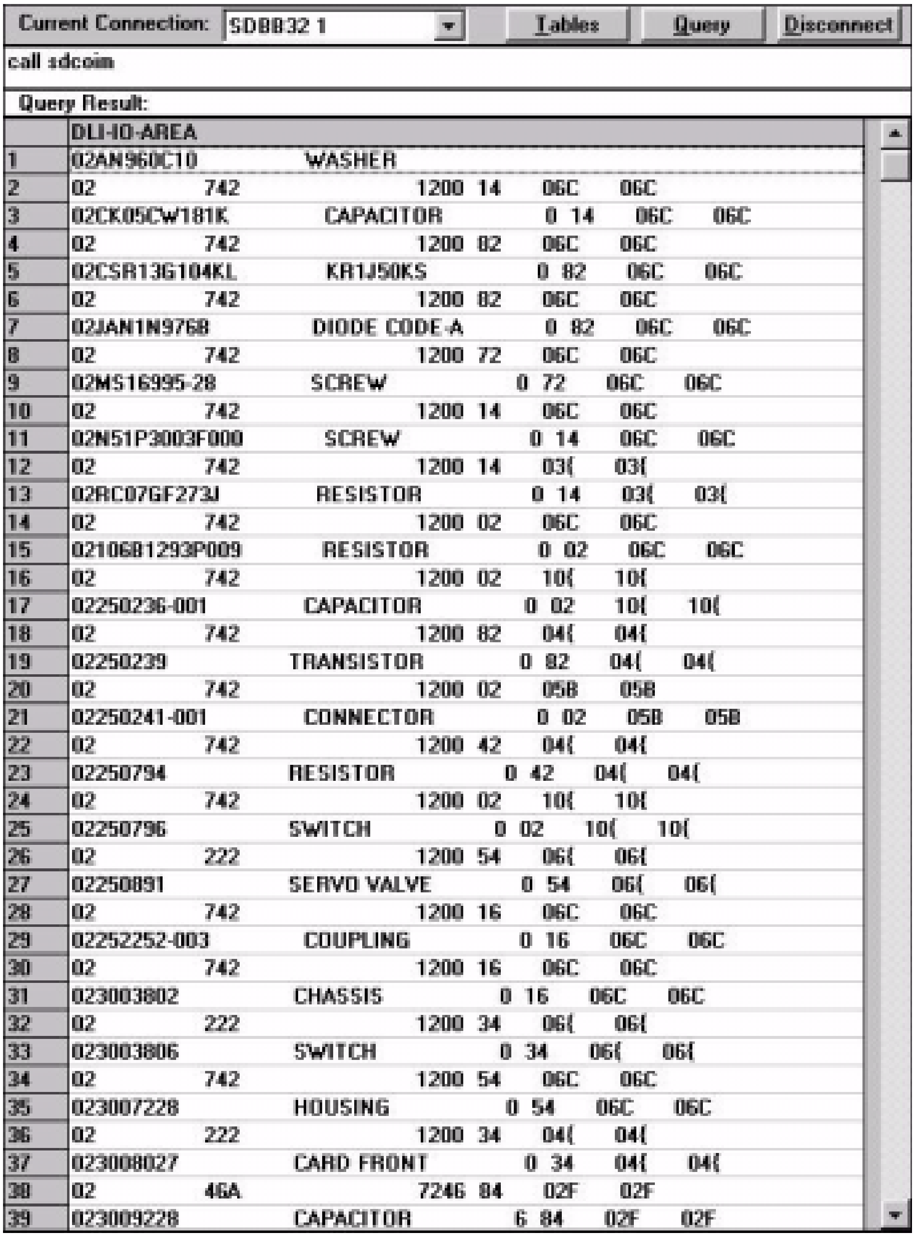
```
Call SDCOIM
```



Call SDCOIM

Figure D–3. VB Demo Call for CCTL/DBCTL

8. Click the <QUERY> button. The data from the display active command appears in the Query Result window as shown in the following screen:



call sdcoim

Query Result:

	DLI-IO-AREA
1	02AN960C10 WASHER
2	02 742 1200 14 06C 06C
3	02CK05CW181K CAPACITOR 0 14 06C 06C
4	02 742 1200 82 06C 06C
5	02CSR13G104KL KR1J50KS 0 82 06C 06C
6	02 742 1200 82 06C 06C
7	02JAN1N976B DIODE CODE-A 0 82 06C 06C
8	02 742 1200 72 06C 06C
9	02MS16995-28 SCREW 0 72 06C 06C
10	02 742 1200 14 06C 06C
11	02N51P3003F000 SCREW 0 14 06C 06C
12	02 742 1200 14 03{ 03{
13	02RC07GF273J RESISTOR 0 14 03{ 03{
14	02 742 1200 02 06C 06C
15	0210681293P009 RESISTOR 0 02 06C 06C
16	02 742 1200 02 10{ 10{
17	02250236-001 CAPACITOR 0 02 10{ 10{
18	02 742 1200 82 04{ 04{
19	02250239 TRANSISTOR 0 82 04{ 04{
20	02 742 1200 02 05B 05B
21	02250241-001 CONNECTOR 0 02 05B 05B
22	02 742 1200 42 04{ 04{
23	02250794 RESISTOR 0 42 04{ 04{
24	02 742 1200 02 10{ 10{
25	02250796 SWITCH 0 02 10{ 10{
26	02 222 1200 54 06{ 06{
27	02250891 SERVO VALVE 0 54 06{ 06{
28	02 742 1200 16 06C 06C
29	02252252-003 COUPLING 0 16 06C 06C
30	02 742 1200 16 06C 06C
31	023003802 CHASSIS 0 16 06C 06C
32	02 222 1200 34 06{ 06{
33	023003806 SWITCH 0 34 06{ 06{
34	02 742 1200 54 06C 06C
35	023007228 HOUSING 0 54 06C 06C
36	02 222 1200 34 04{ 04{
37	023008027 CARD FRONT 0 34 04{ 04{
38	02 46A 7246 84 02F 02F
39	023009228 CAPACITOR 6 84 02F 02F

Figure D–4. Query Results

Shadow OS/390 Web Server

1. Check the syslog of the Shadow control region for the following message after the server has initialized:

```
SWS4353I IMS CCTL SUPPORT ACTIVATED
```

> ***Note:***
> If you do not see the above message on the syslog, do not proceed with the next steps.

2. Run the RPC demo program, `'SWCOIM'.` You must compile the COBOL member, located in the `'NEON.SAMP'` dataset. It is preferable to use LE370 COBOL. If this is not available, use COBOL II. The PART transaction delivered by IBM with IMS is used by the demo program.

3. Enable the following rule for the web IVP:

```
/*WWW /NEON/RPCIMS1 AUTHREQ(NO) SENDTRACE(YES)
/*PROGRAM NAME(SWCOIM) PRELOAD(NO) INVOKE(LINK) TYPE(MODULE) -
         SUBSYS(NONE) PLAN(NONE) PARM() HEADERS(NO)
```

4. Open the web browser and enter the URL

> ***Note:***
> The URL consists of the following:
>
> `http://(DNSname)/(Port Name)/NEON/RPCIMS/`

After you enter the URL, you should see a screen entitled: “IMS SAMPLE PROGRAM USING DBCTL”, followed by data from the IMS PART database. When you see this screen, you will know that the installation was successful.

Transaction Server for IMS Support

The following sections apply to both Shadow Server and Shadow OS/390 Web Server, unless otherwise specified. Information includes:

- Step 1. Modify Installation EXEC.
- Step 2. Modify IMS to activate IMS/APPC.
- Step 3. Create VTAM APPLID.
- Step 4. Modify APPC/MVS.
- Step 5. Install the IMS LU6.2 user edit exit (**optional**).
- Step 6. Verify the IMS Transaction Server installation.
- Step 7. Using the IMS Control Facility.
    a. The APPC/MVS Monitor
    b. The LTERM Facility

Step 1. Modify Installation EXEC

To modify the installation EXEC:

1. Enable the transaction server for IMS by setting up the following parameter using the **MODIFY PARM** command in the IN00:

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

2. Activate the DFSLUEE0 exit (optional for Shadow Server) by using the following **MODIFY PARM** command (see “Step 5. Install the IMS LU 6.2 User Edit Exit (DFSLUEE0)” on page D-15 of this chapter):

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

3. As an option, you can define the following default parameters for the SHADOW_IMS call using the IMS Transaction Server:

   ```
   IF 1 = 1 THEN
      DO

          “MODIFY PARM NAME(IMSCONVTYPE)       VALUE(BASIC)”
          “MODIFY PARM NAME(IMSSYMDEST)        VALUE()”
          “MODIFY PARM NAME(IMSPARTNERLU)      VALUE(P392AIMS)”
          “MODIFY PARM NAME(IMSMODENAME)       VALUE(#BATCH)”
          “MODIFY PARM NAME(IMSRETURNCONTROL)  VALUE(SESSION)”
          “MODIFY PARM NAME(IMSSYNCLEVEL)      VALUE(NONE)”
          “MODIFY PARM NAME(IMSSECURITYTYPE)   VALUE(NONE)”
          “MODIFY PARM NAME(IMSLOCALLU)        VALUE()”

      END
   ```

Where:

> ***Note:***
> Do not modify the IMSCONVTYPE, IMSRETURNCONTROL, or IMSSYNCLEVEL parameters unless instructed by Technical Support.

IMSCONVTYPE

is the conversation type on which the service is invoked There are two possible values, however this value should be set to Basic or omitted altogether:

- **Basic:** (default) TPs will format their data into separate records, with record length and data specified, before sending it.
- **Mapped**: TPs will rely on APPC to format the data that the TPs send.

Do not modify this parameter unless instructed by Technical Support.

IMSSYMDEST

specifies a symbolic name representing the partner LU, the partner 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 data set. 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 eight 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 sym_dest_name parameter value to 8 blanks and provide values for the Partner_LU_name, Mode_name, and TP_name parameters.

IMSPARTNERLU

is the name of the IMS LU as defined in `‘SYS1.PARMLIB(APPCPMxx)’`.

IMSMODENAME

is 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 sym_dest_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 sym_dest_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.

IMSRETURNCONTROL

specifies when the local LU is to return control to the local program, in relation to the allocation of a session for the conversation. Possible values are:

- **SESSION.** (default and recommended value) Specifies to allocate a session for the conversation before returning control to the program. An error in allocating a session is reported on this call.
- **IMMEDIATE.** Specifies to allocate a session for the conversation if a session is immediately available, and return control to the program with a return code indicating whether a session is allocated. An error in allocating a session that is immediately available is reported on this call.
- **CONWINNER.** Specifies to allocate a session in which the local LU is the contention winner, before returning control to the program. As contention winner, the LU avoids having to compete with the partner LU to establish the session, thus potentially saving network traffic. An error in allocating a contention winner session for the conversation is reported on this call.

Do not modify this parameter unless instructed by Technical Support.

IMSSYNCLEVEL

is the synchronization level that the local and partner programs can use on this conversation. The possible values are:

- **NONE.** (default) Program will not perform confirmation processing on this conversation. Programs will not call any services and will not recognize any returned parameters relating to confirmation.
- **CONFIRM.** Programs can perform confirmation processing on this conversation. The programs can call services and will recognize returned parameters relating to confirmation.

Do not modify this parameter unless instructed by Technical Support.

IMSSECURITYTYPE

is the type of security used. Possible values are:

- **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 User_id, Password, and Profile parameters. These values are passed exactly as specified, without folding to uppercase.

IMSLOCALLU

is the name of the local LU from which the caller's allocate request is to originate. The ability to specify the local LU name allows the caller to associate its outbound conversations with particular LUs. The caller's address space must have access to the named LU. Otherwise, a parameter_error return code is returned.

This is the new local LU Name specified in `'SYS1.PARMLIB(APPCPMxx)'` This parameter is optional; the default is to use the APPC Base LU, as is defined in `'SYS1.PARMLIB(APPCPMxx)'`.

Note: It is recommended that a separate Local LU be defined for each Shadow Server you have running using IMS/APPC. Application developers should be informed of which LU to use with which copy of the Shadow Server. *The APPC base LU will work in most cases, however using a separate Local LU tends to be a more reliable request.*

For more information about Shadow_IMS, see the section entitled "SHADOW_IMS" on page D-27 of this chapter.

Note:

If these parameters are not explicitely defined in the call, the system will override any values with the default.

In addition to being changed by the `"MODIFY PARM"` command, these parameters can also be changed dynamically by using **option 5.2** and selecting **PRODAPPCMVS**. For detailed information about these parameters, see the "Started Task Parameters" section located under the PARAMETERS tab of this binder.

Step 2. Modify IMS to activate IMS/APPC

Prerequisites

- Make sure you are using IMS 4.1 or above.
- Make sure that the System parameter in the IMSCTRL macro specifies an MVS version greater than or equal to 4.2, for 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
```

Modification Steps

Before you can use the Transaction Server for IMS, you must first configure IMS/APPC by performing the following steps. Please refer to the *IMS Data Communication Administration Guide* for additional details:

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

Note on IMS/APPC Security:

IMS/APPC security is handled through the RACF resource class TIMS for IMS transaction programs and CIMS for IMS commands. Under ACF/2 and TOP-Secret, if these resource classes are not defined, you will receive a `ATB_SECURITY_NOT_VALID` if attempting to connect to IMS via the IMS Transaction Server. Under ACF/2 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. For example:

```
/SECURE APPC CHECK
```

Step 3. Create VTAM APPLIDs

1. Define the following VTAM APPLIDs:

   - Base APPC APPL
   - IMS APPC APPL
   - Local APPC APPL

Note:
The Base APPC APPL may have already been defined if you have other APPC applications.

```
SDBIMS   VBUILD TYPE=APPL
IMSLU62 APPL ACBNAME=IMSLU62,BASE ACBNAME FOR APPC/IMS     +
   APPC=YES,                                               +
   AUTOSES=0,                                              +
   DDRAINL=NALLOW,                                         +
   DLOGMOD=APPCHOST,                                       +
   DMINWNL=5,                                              +
   DMINWNR=5,                                              +
   DRESPL=NALLOW,                                          +
   DSESLIM=10,                                             +
   LMDENT=19,                                              +
   PARSESS=YES,                                            +
   SECACPT=CONV,                                           +
   SRBEXIT=YES,                                            +
   VPACING=0

MVSLU01 APPL ACBNAME=MVSLU01,BASE ACBNAME FOR APPC         +
   APPC=YES,                                               +
   AUTOSES=0,                                              +
   DDRAINL=NALLOW,                                         +
   DLOGMOD=APPCHOST,                                       +
   DMINWNL=5,                                              +
   DMINWNR=5,                                              +
   DRESPL=NALLOW,                                          +
   DSESLIM=10,                                             +
   LMDENT=19,                                              +
   MODETAB=APPCTAB,                                        +
   PARSESS=YES,                                            +
   SECACPT=CONV,                                           +
   SRBEXIT=YES,                                            +
   VPACING=0                                               +
MVSLU02 APPL   ACBNAME=MVSLU02,LOCAL ACBNAME FOR APPC      +
   APPC=YES,                                               +
   AUTOSES=0,                                              +
   DDRAINL=NALLOW,                                         +
   DLOGMOD=APPCHOST,                                       +
   DMINWNL=5,                                              +
   DMINWNR=5,                                              +
   DRESPL=NALLOW,                                          +
   DSESLIM=10,                                             +
   LMDENT=19,                                              +
   MODETAB=APPCTAB,                                        +
   PARSESS=YES,                                            +
   SECACPT=CONV,                                           +
   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)'`:

```
/************************************************************/
/*                                                          */
/* 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:
These can be added dynamically by putting the define statement `T APPC=xx` in a `SYS1.PARMLIB` member `APPCPMxx` and issuing the `SET APPC=xx` command.

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

Important!
This step is optional for Shadow Server, but is **required** for Shadow OS/390 Web Server using the IMS Auto-HTML feature.

The Shadow OS/390 Web Server and Shadow Server both deliver a load module for the IMS LU 6.2 User Edit Exit. This load module can be found in the load library `'hlq.LOAD(DFSLUEE0)'`.

To install, relink this exit into your IMS `'RESLIB'`, which can be found in the sample library `'hlq.SAMP(JCLLUEE0)'`.

After you have relinked, stop and restart your IMS system for the exit to take effect.

How It Works

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 Shadow OS/390 Web Server or Shadow 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`).

Step 6. Verify the IMS Transaction Server Installation

After you have issued the IMS start command, check the syslog for the following messages to verify that you have a successful IMS Transaction Server install (this section applies to both Shadow Server, and Shadow OS/390 Web Server:

- If the the LUs (logical units) have been successfully added to the APPC, you should see the following message on the syslog:

```
ATB050I LOGICAL UNIT IMSLU62 FOR TRANSACTION SCHEDULER  IVP1
HAS BEEN ADDED TO THE APPC CONFIGURATION.
```

- If IMS has successfully been connected to the APPC/MVS, you should see the following message on the syslog and the jeslog of the IMS control region:

```
DFS1960I IMS HAS REQUESTED A CONNECTION WITH APPC/MVS   IVP1
DFS1958I IMS CONNECTION TO APPC/MVS COMPLETE, LUNAME=xxxxxxxx
   IVP1
```

For Shadow Direct:

Submit a transaction request from NEON Client:

> ***Note:***
> Steps 1 through 4 apply to both generic and customized IMS RPCs. Steps 5 and 6 apply to generic only, and steps 7 and 8 apply to customized only.

Generic and Custom RPCs:

1. Configure an ODBC data source. For further information, see the Shadow Server User’s Guide chapter, “Configuring Data Sources”.

2. Do one of the following, depending on the operating system of the client PC:

   – **Windows 95/NT.** From the client PC Start menu, select **Start**>**Programs**>**Shadow Direct**>**VB Demo Application**. The VB Demo window opens.

   – **Windows 3.x.** Select the Shadow group icon, then double-click the VB Demo application icon. The VB Demo window opens.

3. In the VB Demo window, select **Connections**>**Add New Connection** (see Figure D–1). The Select Data Source window opens (see Figure D–2).

4. Select a data source using either the File Data Source or Machine Data Source tab, then click OK.

> ***Note:***
> If you are using a generic RPC, go to step 5. If you are using a customized RPC, go to step 7.

*Generic RPC **Only***

5. In the input box (the white area directly below the Current Connection listing), type the following:

```
Call SHADOW_IMS('IMS','/DISPLAY
','P392AIMS','NONE','ACTIVE','APPC-IMS-DATA')
```

For more information about Shadow_IMS, see the section entitled "SHADOW_IMS" on page D-27 of this chapter.

Note:
In the above statement, `P392AIMS` should be relaced with whatever IMS/APPC LUNAME you defined in the IMS APPC APPLID, see "Step 3. Create VTAM APPLIDs" on page D-13. In the example on page D-4, it would be `IMSLU62`.

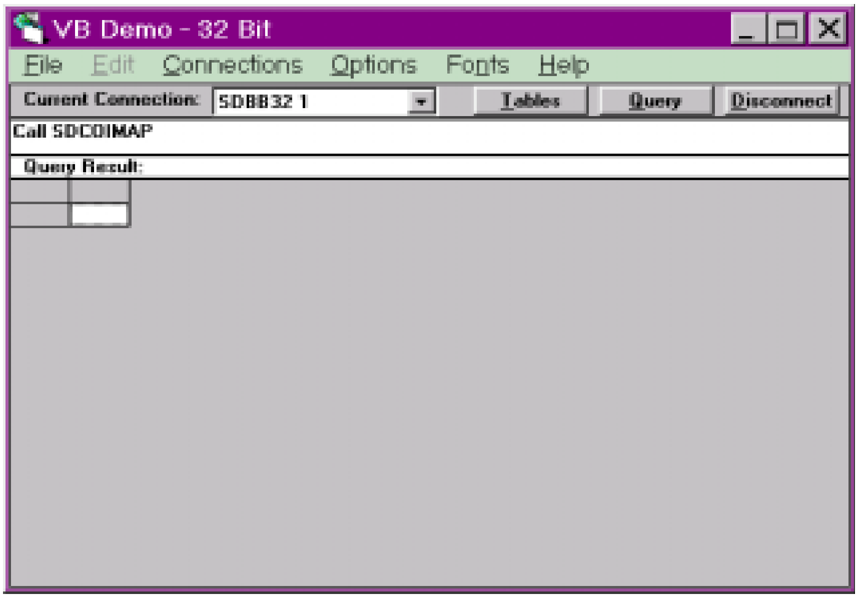
6. Click the <QUERY> button. The data from the display active command appears in the Query Result window (see Figure D–4).

Custom RPC **Only**

7. To run the RPC demo program, `'SDCOIMAP'`, you must compile the COBOL member, located in the `'NEON.SAMP'` file. It is preferable to use COBOL for MVS, followed in preference by LE370 COBOL. If neither of these is available, use COBOL II. The PART transaction delivered by IBM with IMS is used by the demo program.

8. In the input box, type the following:

```
CALL SDCOIMAP
```



SDBB32 1

Call SDCOIMAP

Figure D–5. VB Demo Call for IMS Transaction Server

9. Click the <QUERY> button. The data from the customized RPC appears in the Query Result window (see Figure D–4).

For Shadow OS/390 Web Server

Web Browser

1. Edit the following rule to reflect the correct LU name:

```
/*WWW /TEST/IMSEXEC0
**************************************************************
*                                                            *
* SAMPLE APPLICATION WHICH ILLUSTRATES THE USE OF AN EXECSQL *
* PROCESS SECTION.  THE AUTOFORMAT KEYWORD CALLS FOR THE     *
* ROW DATA TO BE FORMATTED INTO AN HTML TABLE.               *
*                                                            *
**************************************************************

/*EXECSQL MAXROWS(100) -
      AUTOFORMAT( TITLE('SAMPLE IMS QUERY USING /*EXECSQL') -
                  BODY('BGCOLOR="#FFCC33"') -
                )
CALL SHADOW_IMS('IMS','/DIS','P392AIMS','NONE','APPC','APPC-
IMS-DATA')
```

   For more information about Shadow_IMS, see the section entitled “SHADOW_IMS” on page D-27 of this chapter.

2. Enable the rule.

3. Open the web browser.

4. Enter the URL. The browser should present the results of the command.

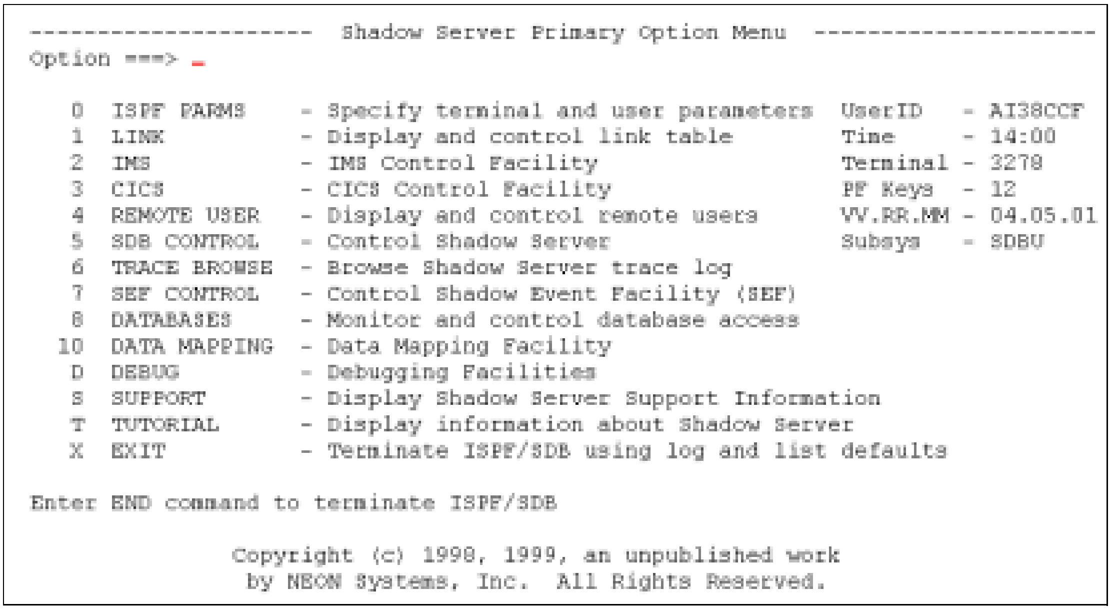
Step 7. Using the IMS Control Facility

The Shadow Server IMS Control Facility allows the user to monitor and control access to IMS/TM. This facility consists of the following:

- APPC/MVS Monitor for monitoring APPC/MVS conversations to IMS.
- LTERM Facility for exploiting existing IMS LTERM security by assigning known LTERM names to inbound IMS transactions based upon the userid or IP address of the originating requestor.

Note:
Both of these features are **optional** for Shadow Server and Shadow OS/390 Web Server.

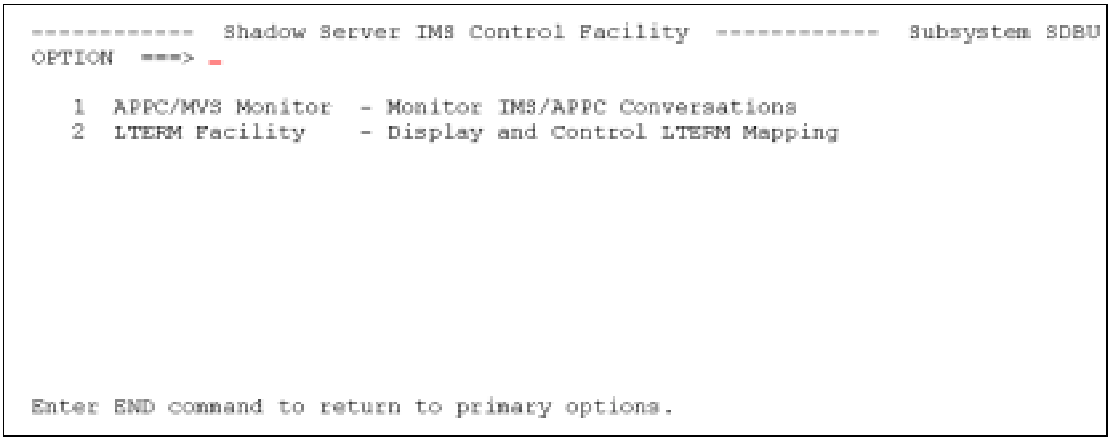
The IMS Control Facility can be accessed by selecting the IMS option from the Primary Option ISPF panel for either Shadow Server or Shadow OS/390 Web Server.



```
---------------------  Shadow Server Primary Option Menu  ---------------------
Option ===> _

   0  ISPF PARMS    - Specify terminal and user parameters  UserID   - AI38CCF
   1  LINK          - Display and control link table        Time     - 14:00
   2  IMS           - IMS Control Facility                  Terminal - 3278
   3  CICS          - CICS Control Facility                 PF Keys  - 12
   4  REMOTE USER   - Display and control remote users      VV.RR.MM - 04.05.01
   5  SDB CONTROL   - Control Shadow Server                 Subsys   - SDBU
   6  TRACE BROWSE  - Browse Shadow Server trace log
   7  SEF CONTROL   - Control Shadow Event Facility (SEF)
   8  DATABASES     - Monitor and control database access
  10  DATA MAPPING  - Data Mapping Facility
   D  DEBUG         - Debugging Facilities
   S  SUPPORT       - Display Shadow Server Support Information
   T  TUTORIAL      - Display information about Shadow Server
   X  EXIT          - Terminate ISPF/SDB using log and list defaults

Enter END command to terminate ISPF/SDB

               Copyright (c) 1998, 1999, an unpublished work
                by NEON Systems, Inc.  All Rights Reserved.
```

Figure D–6. Shadow Primary Options Menu

This will take you to the Shadow Server IMS Control Facility panel, from which you can select either the APPC/MVS Monitor or the LTERM Facility:



```
------------  Shadow Server IMS Control Facility  ------------  Subsystem SDBU
OPTION  ===> _

   1  APPC/MVS Monitor   - Monitor IMS/APPC Conversations
   2  LTERM Facility     - Display and Control LTERM Mapping











Enter END command to return to primary options.
```

Figure D–7. Shadow IMS Control Facility Panel

The APPC/MVS Monitor

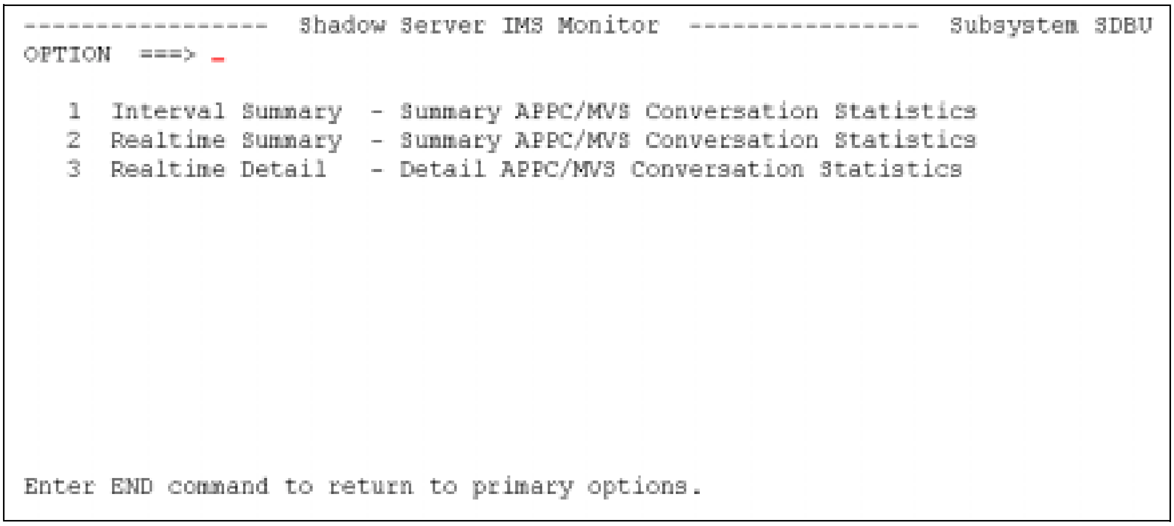
Note:
This monitor can be used with Transaction Server for IMS **only**. It **cannot** be used for any other Transaction Server or any other IMS support.

The Shadow Server APPC/MVS Monitor allows you to monitor and control your APPC/MVS conversations with IMS/TM, in real-time and historical mode. APPC/MVS conversations can be terminated automatically by means of inactivity timeout settings, or manually by means of a line command.

Activating and Using the APPC/MVS Monitor

To activate the APPC/MVS Montor:

1. Set the MONITORAPPC/MVS parameter to YES.
2. Select option 1 from the panel shown in Figure D–7 to access the Shadow Server IMS Monitor panel:



```
-----------------  Shadow Server IMS Monitor  ----------------  Subsystem SDBU
OPTION  ===> _

   1  Interval Summary  - Summary APPC/MVS Conversation Statistics
   2  Realtime Summary  - Summary APPC/MVS Conversation Statistics
   3  Realtime Detail   - Detail APPC/MVS Conversation Statistics










Enter END command to return to primary options.
```

Figure D–8. Shadow Server IMS Monitor panel

From this panel you can monitor statistics for:

- Interval summary APPC/MVS conversations
- Realtime summary APPC/MVS conversations
- Detail APPC/MVS conversations

3. Select options 1 and 2 to view the following information for interval summary and realtime summary APPC/MVS conversations:

- The start time for the interval.
- The total number of conversations.
- The total number of allocated conversations.
- The total number of sends.
- The total number of receives.
- The total number of active conversations.
- The total amount of data sent.
- The total amount of data received.

4. Select option 3 to view the following detailed APPC/MVS conversation statistics:

   - The TP name, or name of the transaction to be executed.
   - The conversation start time.
   - The time in which the last APPC/MVS call was issued.
   - The last APPC/MVS call return code.
   - The last APPC/MVS call reason code.

Setting the APPC/MVS Monitor Parameters

The following parameters apply to the control of the APPC/MVS Monitor:

MONITORAPPC/MVS
specifies whether or not to monitor APPC/MVS conversations. This parameter should be set to YES in order to activate the monitor.

IMSDEALLOCONVTIME
specifies the the maximum allowable duraction of activity for any conversation. The inactive period is defined as the time expired since the last APPC/MVS call.

CHECKCONVIDINTERVAL
controls how often each convid is checked to see if the convid has timed out. If the convid has timed out, the conversation is deallocated and the entry in the convesation id table is removed.

IMSCNVIDTBLSZ
is used to specify the size of the table used to maintain the status of active conversations.

REALTIMESUMMARY
controls the number of APPC/MVS realtime summary records to keep in memory at one time. If this parameter is set to zero, no APPC/MVS realtime summary records will be retained in memory.

There are also parameters to control the tracing and logging for the monitor. These include:

TRACEAPPCMVSMN
controls whether or not the APPC/MVS monitor data collection APIs are to be traced. This parameter should only be turned on if the monitor is not functioning correctly.

LOGAPPC/MVSSUM
controls whether or not APPC/MVS interval summary information should be logged.

LOGAPPC/MVSSUMTABLE
is used to set the name of the DB2 table used to log APPC/MVS interval summary information.

LOGRETAINAPMVSSUM
controls the number of days to wait before automatically deleting rows from the APPC/MVS summary table.

All APPC/MVS Monitor parameters can be changed dynamically by using **option 5.2** and selecting the applicable parameter group:

- PRODAPPC/MVS for the general APPC/MVS Monitor parameters.
- PRODTRACE for the APPC/MVS Monitor tracing parameters.
- PRODLOGGING for the APPC/MVS Monitor logging parameters.

They can also be changed using the **MODIFY PARM** command in the `IN00` as the following example shows:

```
“MODIFY PARM NAME(MONITORAPPC/MVS)     VALUE(NO)”
“MODIFY PARM NAME(IMSDEALLOCONVTIME)   VALUE(900 SECONDS)”
“MODIFY PARM NAME(CHECKCONVIDINTERVAL) VALUE(15 SECONDS)”
“MODIFY PARM NAME(IMSCNVIDTBLSZ)       VALUE(32K)”
“MODIFY PARM NAME(REALTIMESUMMARY)     VALUE(60 INTERVALS)”
“MODIFY PARM NAME(TRACEAPPCMVSMN)      VALUE(NO)
“MODIFY PARM NAME(LOGAPPC/MVSSUM)      VALUE(NO)
“MODIFY PARM NAME(LOGAPPC/MVSSUMTABLE) VALUE(SHADOW.
                                       APMVSSUM)
“MODIFY PARM NAME(LOGRETAINAPMVSSUM)   VALUE(0 DAYS)
```

For detailed information about these parameters, see the “Started Task Parameters” section located under the PARAMETERS tab of this binder.

The LTERM Facility

The LTERM FAcility provides the capability of exploting existing IMS LTERM security by assigning known LTERM names to inbound IMS transactions based on either the userid or the IP address of the originating requestor.

Assign LTERMs to IMS Transactions

Note:
This step is optional for both Shadow Server and Shadow OS/390 Web Server.

IMS provides the application program with an I/O PCB containing information about with whom the application program is interacting. This control block contains the LTERM (Logical Terminal) name of the terminal, Userid of the person who signed on, and the MODNAME of the input message. NEON automatically updates the Userid and MODNAME information. Unfortunately, a distributed IP network does not provide a mechanism for capturing an LTERM-equivalent name, but with this feature, you can now assign meaningful LTERM names based upon Userid's or TCP/IP Addresses.

When an IMS transaction is passed to Shadow OS/390 Web Server, the IMS Transaction Server scans a table looking for a matching Userid or TCP/IP address provided the IMSLTERMTABLESEQ parameter (PRODIMS group) is set to USERID or IPADDRESS.

Note:
Even with IMS LTERM Table entries present, the parameter must be set to USERID or IPADDRESS, otherwise a scan is not performed. The default for the IMSLTERMTABLESEQ parameter is NONE.

If a matching entry is found, the associated LTERM is passed to the NEON-Supplied IMS Exit (`DFSLUEE0`), which places the entry into the I/O PCB for use by the IMS application program.

The IMS LTERM Table is maintained in Extended Private storage within the server's address space. Entries in the table can contain generic Userids and TCP/IP addresses. Once added, these entries can be enabled or disabled at any time. The changes are effective immediately.

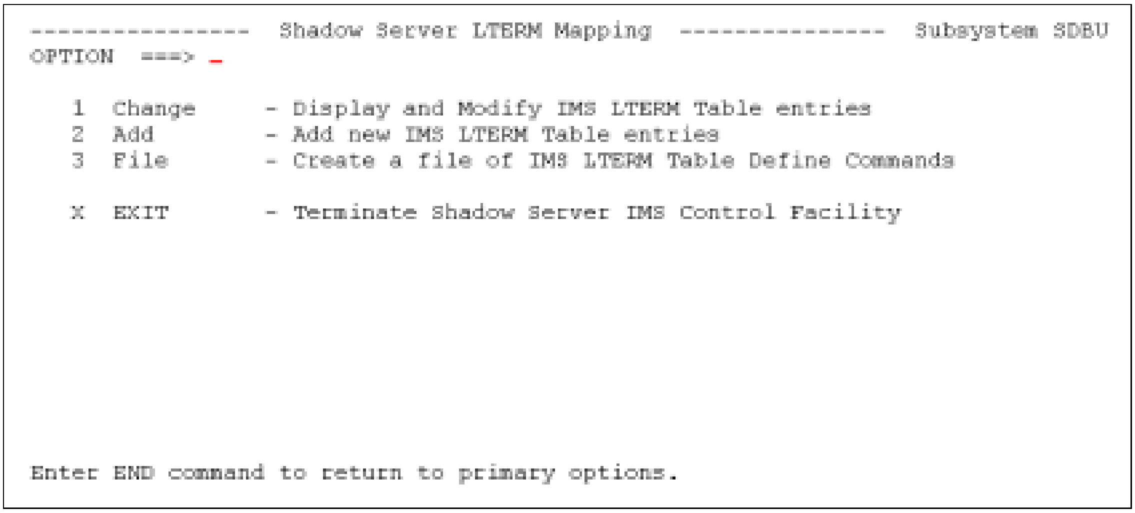
Many IMS application programs use the LTERM to determine things such as output printer destination and application-based security.

Note:
Help screens are available by pressing <PF1>.

Access the LTERM Facility

The LTERM facility is located in the IMS Control Facility (see Figure D–7). To acccess:

1. Select option 2, LTERM Facility, to display and control LTERM mapping. This will take you to the LTERM Mapping Panel:



```
----------------  Shadow Server LTERM Mapping  ---------------  Subsystem SDBU
OPTION  ===> _

   1  Change     - Display and Modify IMS LTERM Table entries
   2  Add        - Add new IMS LTERM Table entries
   3  File       - Create a file of IMS LTERM Table Define Commands

   X  EXIT       - Terminate Shadow Server IMS Control Facility









Enter END command to return to primary options.
```

Figure D–9. Shadow LTERM Mapping Panel

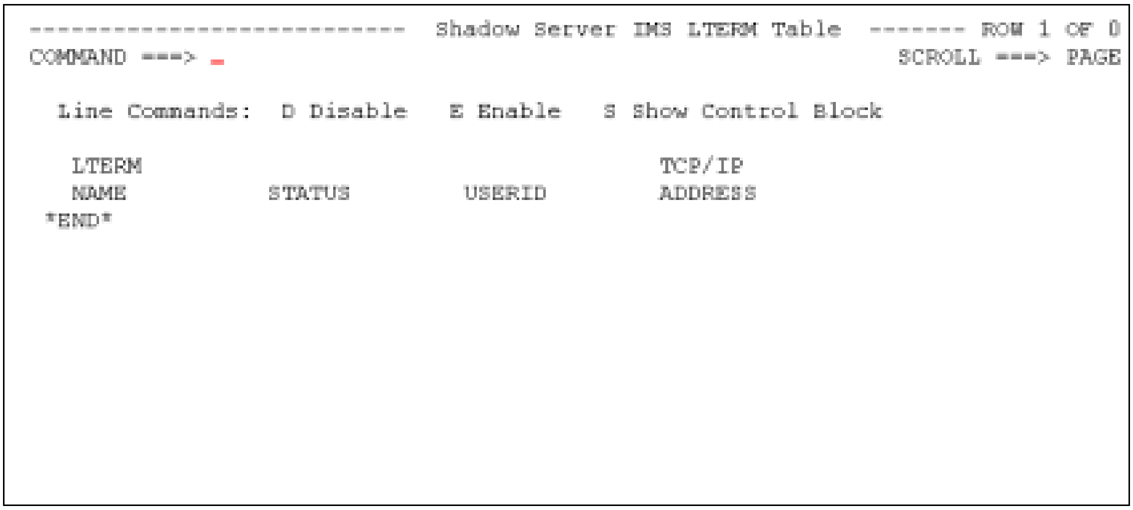
This panel allows you to choose from one of the following options:

- Display and modify IMS LTERM Table entries.
- Add new IMS LTERM Table entries
- Create a file of IMS LTERM Table Define Commands.

Note:

IMS LTERM Table entries are added, modified and displayed using **DEFINE, MODIFY** and **DISPLAY** commands. These commands can be entered via a REXX exec (under ADDRESS SDB or SWS) such as `‘SDBBIN00’` or `‘SWSSIN00’`.

2. Select option 1 from the LTERM Mapping panel to display and modify IMS LTERM table entries. This will access the following IMS LTERM Table panel:



```
--------------------------  Shadow Server IMS LTERM Table  ------- ROW 1 OF 0
COMMAND ===> _                                                 SCROLL ===> PAGE

  Line Commands:  D Disable   E Enable   S Show Control Block

   LTERM                                       TCP/IP
   NAME          STATUS        USERID          ADDRESS
 *END*
```

Figure D–10. Shadow IMS LTERM Table Panel

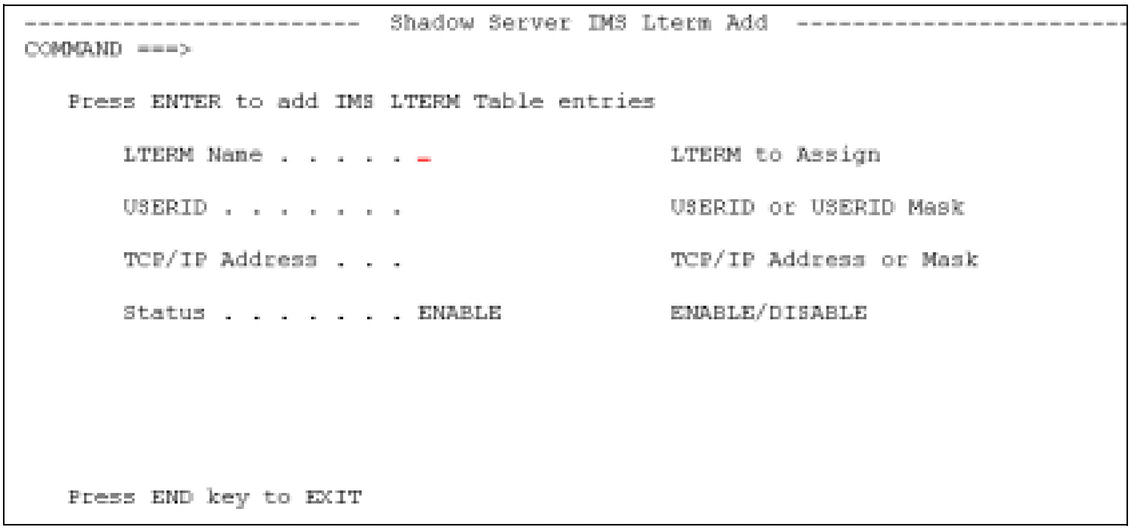
This panel provides a list of all the table entries in the server, which is initially displayed in LTERM sequence. This list can be sorted into Userid, TCP/IP Address and LTERM sequence with a SORT command followed by a sequence parameter of USERID, IPADDR or LTERM.

3. If necessary, modify any IMS LTERM table entries:

   a. Overtype the LTERM NAME field in the IMS LTERM Table panel.

   b. Press <ENTER>. Multiple entries can be modified before pressing <ENTER>.

4. Select option 2 from the LTERM Mapping panel (Figure D–9) to add a new LTERM table entry. This will access the following IMS LTERM Add panel:

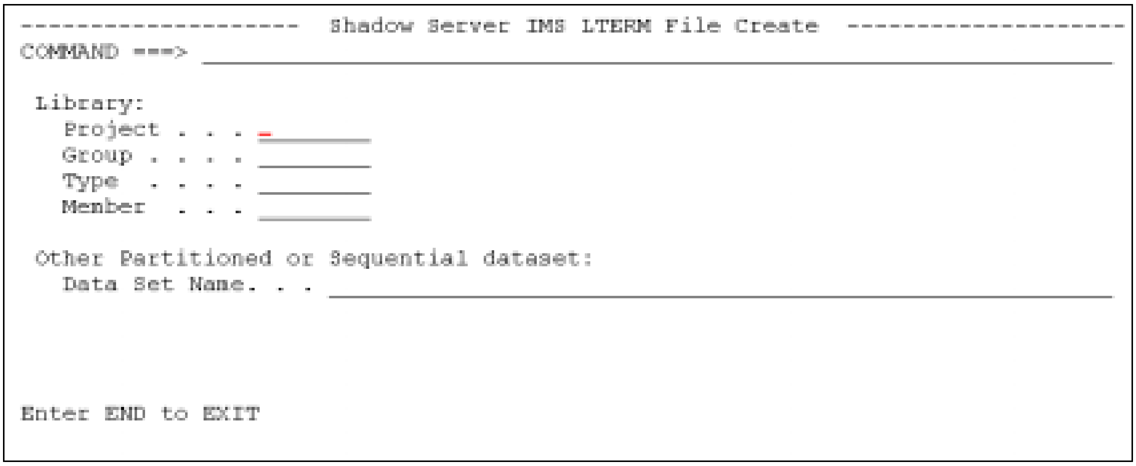


```
------------------------  Shadow Server IMS Lterm Add  ------------------------
COMMAND ===>

   Press ENTER to add IMS LTERM Table entries

       LTERM Name . . . . . _                  LTERM to Assign

       USERID . . . . . . .                    USERID or USERID Mask

       TCP/IP Address . . .                    TCP/IP Address or Mask

       Status . . . . . . . ENABLE             ENABLE/DISABLE




   Press END key to EXIT
```

Figure D–11. Shadow IMS LTERM Add Panel

The IMS Control Facility Add screen provides the ability to add new entries to the IMS LTERM table in the server. These are dynamic in nature and disappear when the server is “bounced”. The only permanent entries are those added through **DEFINE** commands in the `‘SWSSIN00’` or `‘SDBBIN00’` specification.

5. Select option 3 from the LTERM Mapping panel (Figure D–9) to create a file of the current IMS LTERM table. This will acess the following the IMS LTERM File Create panel:



```
--------------------  Shadow Server IMS LTERM File Create  --------------------
COMMAND ===> _________________________________________________________________

 Library:
   Project . . . _______
   Group . . . . ________
   Type  . . . . ________
   Member  . . . ________

 Other Partitioned or Sequential dataset:
   Data Set Name. . . ________________________________________________________



Enter END to EXIT
```

Figure D–12. Shadow IMS LTERM File Create Panel

The IMS Control Facility File screen, provides the ability to create a file of **DEFINE** commands for all the IMS LTERM Table entries currently stored in the server. This allows users to add table entries and create a file that can be added to, or run from, `‘SWSSIN00’` or `‘SDBBIN00’`.

Format

The format of the commands are:

```
DEFINE IMSLTERM USERID(xxxxxxxx) IPADDRESS(xxxxxxxxxxxxxxx)
LTERM(xxxxxxxx) STATUS(ENABLE)

MODIFY IMSLTERM USERID(xxxxxxxx) IPADDRESS(xxxxxxxxxxxxxxx)
LTERM(xxxxxxxx) STATUS(DISABLE)

DISPLAY IMSLTERM USERID(xxxxxxxx) or DISPLAY IMSLTERM
IPADDRESS(xxxxxxxxxxxxxxx) STATUS(ENABLE)
```

SHADOW_IMS

A generic RPC is provided that will allow you to invoke an existing transaction. This RPC is called SHADOW_IMS for Shadow Server, and can be invoked in the following ways:

- For Shadow Server, from any ODBC-compliant application.
- For Shadow OS/390 Web Server, from an /*EXECIMS rule or an /*EXECSQL rule on the client workstation as a pass-through query (please see the Visual Basic and PowerBuilder samples).

Shadow Server

For Shadow Server, the SHADOW_IMS RPC is invoked using the following ODBC CALL statement:

```
CALL SHADOW_IMS(’IME’,’TRANSACTION PROGRAM NAME’,’IMS-PARTNER-LU
NAME’,’SECURITY-TYPE’,’TP PARAMETERS’,’COLUMN NAME’,’LOCAL LU
NAME’,’MODE NAME’,’SYMDEST’,’USERID’,’PASSWORD’,’PROFILE’,’SEND-
TYPE’,’MESSAGE-LENGTH’)
```

Where:

IME

is a keyword describing the interface.

TRANSACTION PROGRAM NAME

is the name of the IMS transaction to be executed.

IMS-PARTNER-LU-NAME

is the name of the IMS Partner LU as defined in ‘SYS1.PARMLIB(APPCPMxx)’.

SECURITY TYPE

defines the type of security in use:

- **NONE** specifies to omit access security information on this allocation request.

- **SAME** specifies to 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** specifies to use the access security information that the local program provides on the call. The local program provides the information by means of the User_id, Password, and Profile parameters. These values are passed exactly as specified, without folding to uppercase.

TP PARAMETERS

are the parameters for Transaction Program.

COLUMN NAME

is the column name or map name used for returned data. For the `MAP` keyword, the syntax is: `‘MAP(NAME(PARTREXX) FIELDS(*))’`

- **NAME:** This entry should correspond to the name assigned to the map during extraction.

- **FIELDS:** There are two ways to return data from all columns that are enabled in the map definition:
  - Use an asterisk after FIELDS.
  - Omit FIELDS altogether

To exclude some columns, enter the names of the enabled columns you want returned in parentheses after FIELDS.

For more information about the Data Mapping facility, refer to the appropriate chapter in the Shadow *User's Guide*.

LOCAL LU NAME

specifies the name of the local LU Name (default `'P392AIM1'`) from which the caller's allocate request is to originate. The ability to specify the local LU name allows the caller to associate its outbound conversations with particular LUs. The caller's address space must have access to the named LU. Otherwise, a parameter_error return code is returned.

This is the new local LU Name specified in `‘SYS1.PARMLIB(APPCPMxx)’` This parameter is optional; the default is to use the APPC Base LU, as is defined in `‘SYS1.PARMLIB(APPCPMxx)’`.

Note: It is recommended that a separate Local LU be defined for each Shadow Server you have running using IMS/APPC. Application developers should be informed of which LU should be used with which copy of the Shadow Server. *The APPC base LU will work in* *most cases, however using a separate Local LU tends to be more reliable.*

MODE NAME

specifies the mode name 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 sym_dest_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 sym_dest_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.

SYMDEST

specifies a symbolic name representing the partner LU, the partner 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 data set. 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 eight characters, it must be left-justified in the variable field, and padded on the right with blanks. To not specify a symbolic destination name, set the sym_dest_name parameter value to 8 blanks and provide values for the Partner_LU_name, Mode_name, and TP_name parameters.

USERID

specifies the userid. The partner LU uses this value and the password to verify the identity of the end user that initiated the allocation request. The partner LU may use this value for auditing and accounting purposes, and, together with the security profile (if present), to determine which partner programs the local program can access.

When the partner LU is on MVS with RACF protection, the userid must be 1-8 alphanumeric characters.

This parameter is significant only when the Security_type parameter contains a value of Pgm. Otherwise, this parameter has no meaning and is ignored.

PASSWORD

specifies the password. The partner LU uses this value and the userid to verify the identity of the end user that made the allocation request. When the partner LU is on MVS with RACF protection, the password must be 1-8 alphanumeric characters padded with blanks.

This parameter is significant only when the Security_type parameter contains a value of Pgm. Otherwise, this parameter has no meaning and is ignored.

PROFILE

specifies additional security information that may be used to determine what partner programs the local program may access, and which resources the local program may access. When the partner LU is on MVS with RACF protection, APPC/MVS treats the profile name as a RACF group name for verifying access to partner programs. The profile name must be 1-8 alphanumeric characters.

This parameter is significant only when the Security_type parameter contains a value of Pgm. Otherwise, this parameter has no meaning and is ignored.

SEND-TYPE

specifies what, if any, information is to be sent to the partner program in addition to the data supplied. Send_type also lets you combine operations (for example, Send_and_confirm) and save extra calls to APPC. Default value is 1. Valid values for this parameter are:

- **0 Buffer_data**
  Specifies that no additional information is to be sent to the partner program, and the data may be buffered until a sufficient quantity is accumulated.
- **1 Send_and_flush**
  Specifies that no additional information is to be sent to the partner program. However, the supplied data is sent immediately rather than buffered. This is functionally equivalent to a Send_data call with the Send_type parameter set to Buffer_data followed by a Flush call.
- **2 Send_and_confirm**
  Specifies that the supplied data is to be sent to the partner program immediately, along with a request for confirmation. This is functionally equivalent to a Send_data call with the Send_type parameter set to Buffer_data followed by a Confirm call.

- **3 Send_and_prepare_to_receive**
  Specifies that the supplied data is to be sent to the partner program immediately, along with send control of the conversation. This is functionally equivalent to a Send_data call with the Send_type parameter set to Buffer_data followed by a Prepare_to_receive call with the prepare_to_receive_type set to sync_level and the locks parameter set to short.
- **4 Send_and_deallocate**
  Specifies that the supplied data is to be sent to the partner program immediately, along with a deallocation notification. This is functionally equivalent to a Send_data call with the Send_type parameter set to Buffer_data followed by a Deallocate call with the deallocate_type set to sync_level.

MESSAGE-LENGTH

specifies the length of the messages that are written to or read from the message queue. Default value is 32k.

APPENDIX E:
Transaction Server for CICS

This appendix covers the following topics for Transaction Server for CICS:

- Installation
- Shadow CICS RPC
- CICS Load Balancing
- EXCI Failover

See Chapter 6 of the *Shadow Programming Guide* for more information about Transaction Server for CICS.

Installation

CICS Requirements

1. CICS server region must be CICS 4.1 or higher. However, you can ‘daisy chain’ from initial CICS server region to lower level CICS region by allowing the CICS 4.1 system to route the request to a lower level system.

2. DFHIRP module in LPA must be at CICS 4.1 level or higher.

3. DFHCSVC module in LPA must be at CICS 4.1 level or higher.

4. If you are running MVS/ESA V4.3 or lower, the Shadow Direct/Shadow OS/390 Web Server started task must execute in the same MVS image as CICS server region.

5. If you are running MVS/ESA V5.1 or higher, the Shadow Direct/Shadow OS/390 Web Server started task and CICS server region must be in the same MVS sysplex.

Modify Started Task

1. To activate the Transaction Server for CICS, make the `‘SDFHAUTH’` library available to the Shadow OS/390 Web Server address space via the `‘LNKLSTxx’` member in `‘SYS1.PARMLIB’` or the STEPLIB DD statement. If this library is used in the STEPLIB concatenation, it must be APF authorized.

2. Make the `‘SDFHLOAD’` library available to the Shadow Direct/Shadow OS/390 Web Server address space via the `‘LNKLSTxx’` member in `‘SYS1.PARMLIB’` or the by the `‘SDBRPLLB’` DD statement.

3. To activate the External CICS Interface (EXCI) in Transaction Server for CICS, you must add the `‘SDBRPLLB’` DD statement to the started task procedure using the DSN of the library that contains the EXCI modules. The DSN that is supplied by CICS is “CICS410.SDFHEXCI”.

Modify Initialization EXEC

Create the parameters for the Transaction Server for CICS in the initialization EXEC dataset in `‘SDBxIN00’` (Shadow Server) and `‘SWSxIN00’` (Shadow OS/390 Web Server) in the EXEC library. The parameters for the server are listed below.

1. Set the CICS subsystem name in `‘IEFSSNxx’`. If omitted, the default value is “CICS”.

   `“MODIFY PARM NAME(CICSSUBSYSTEM) VALUE(CICS)”`

2. Set the CICS svc number in `‘IEASVCxx’`. If omitted, the default value is “216”.

   `“MODIFY PARM NAME(CICSIRCSVCNO) VALUE(216)”`

3. Set the name of Transaction Server for CICS. If this parameter is omitted, the default value is the SSID of Shadow Direct/Shadow OS/390 Web Server.

   `“MODIFY PARM NAME(CICSTXNSERVERNAME) VALUE(SDBB)”`

4. Enable the EXCI interface (valid only for CICS 4.1 and above).

   `“MODIFY PARM NAME(EXCI) VALUE(YES)”`

5. Create a statement to define the connection. This statement consists of the following parameters:

   `“DEFINE CONNECTION”`
   Required. Define the connection to CICS.

   `“NAME(EXCS)”`
   Required. Connection name for CICS Txn Server. Name must be unique.

   `“GROUP(DFH$EXCI)”`
   Required. Group name for CICS Txn Server. Every resource must have a group name, and name must be unique.

   `“ACCESSMETHOD(IRC)”`
   Required. Access method used on this link. MUST match access method in CICS connection.

   `“NETNAME(BATCHCLI)”`
   Required. Network name identifying remote system. Must match netname in CICS connection definition.

   `“SECURITYNAME(EXCS)”`
   Optional. If present, must be a valid security name of remote system.

`“INSERVICE(YES)”`
Required. Status of connection.

`“PROTOCOL(EXCI)”`
Required. Type of protocol used on link. Must match protocol in CICS connection.

`“APPLID(A06CICS1)”`
Required. VTAM applid of target CICS.

`“ALTAPPLID(A06CICS2)”`
Optional. VTAM applid of alternate CICS.

6. Create a statement to define the session. This statement consists of the following parameters:

`“DEFINE SESSION”,`
Define the session associated to the connection to CICS.

`“NAME(EXCS)”,`
Required. Session name for CICS Txn Server.

`“GROUP(DFH$EXCI)”,`
Required. Group name for CICS Txn Server (must match the connection group parameter).

`“CONNECTION(EXCS)”,`
Required. Name of connection definition to be used with this session.

`“PROTOCOL(EXCI)”`
Required. Type of protocol to be used for CICS link.

`“RECEIVEPFX()”,`
Ignored. Prefix used as the first characters of the receive session name.

`“RECEIVECOUNT()”,`
Ignored. Number of parallel sessions that are usually received before sending. Blank/zero for EXCI connections.

`“SENDPFX(SN)”,`
Required. Prefix used as the first characters of the send session name.

`“SENDCOUNT(4)”,`
Required. Number of parallel sessions that are usually sent before receiving.

Note:
The `SENDCOUNT()` has to be less than or equal to the `RECEIVECOUNT()` in your CICS session definition. Also, the TOTAL number of all your `SENDCOUNT()` in your exec can NOT exceed 25. This is a limitation of CICS EXCI Interface due to the fact that each EXCI client can not have more than 25 MRO logons. If you use CICS 5.2 or above, this limit is 100. This limit can be increased by applying a usermod provided by IBM; contact your IBM account representative about obtaining this usermod.

`“IOAREALEN(4096)”`
Required. Length, in bytes, of the terminal input/output area to be used for processing messages transmitted on the link.

7. If you want to have connections to more than one CICS, simply repeat step 5 and 6. Remember that the `NAME()` parameter (connection name for your CICS Txn Server) in your EXEC has to be unique.

8. Recommended high performance options for the Transaction Server for CICS:

   - `MODIFY PARM NAME (EXCIPIPEPREALLOC) VALUE(YES)`
   - `MODIFY PARM NAME (EXCIPIPEPREOPEN) VALUE(YES)`

Configure CICS

1. Verify that CICS Multi-Region Operation (MRO) is installed before continuing. Please refer to the CICS Installation Guide for instructions on installing MRO.

2. If IRC is not present and open, the server will not be able to successfully connect to CICS. IRC must be defined, installed, and started. To verify that IRC is up and running in CICS, you can use the “**CEMT I IRC**” command. The response from the command should appear as follows:

```
I IRC
STATUS: RESULTS - OVERTYPE TO MODIFY
Irc Ope
```

3. For the demo:

   a. Define the sample VSAM file supplied by IBM with CICS, FILEA.
   b. Load the test data to the FILEA file.
   c. Insert the FILEA DD statement in your CICS startup procedure.
   d. Install the group containing the FILEA definitions, named `DFH$FILA`:
   `CEDA INSTALL GROUP(DFH$FILA)`
   e. Install the sample EXCI support group supplied by IBM for CICS, named `DFH$EXCI`:
   `CEDA INSTALL GROUP(DFH$EXCI)`

4. Use the members named EXCS in the `DFH$EXCI` group as examples when creating site-specific connection and session definitions.

Note:
The connection type has to be specific.

CICS Security

Because Shadow Server utilizes the External CICS Interface (EXCI) to access CICS programs, and EXCI uses the CICS MRO, EXCI is subject to the same security checks as a CICS system connecting to another CICS system using MRO.

- **MRO LOGON Security**

  This security check will ensure that the Shadow Direct/Shadow OS/390 Web Server started task is allowed to use a specific MRO connection. This is done when a pipe is allocated during MRO logon.

  If you start your Shadow Direct/Shadow OS/390 Web Server started task, and the `ALLOCATE_ PIPE` request fails with `RESPONSE(SYSTEM_ ERROR)`, `REASON(IRC_ LOGON_ FAILURE)`, and `subreason1= 204(decimal)`, you will need to set up MRO LOGON security for Shadow Direct/Shadow OS/390 Web Server. The Shadow Direct/Shadow OS/390 Web Server's userid must be authorized with **UPDATE** authority to `DFHAPPL.net_name` RACF profile, where net_ name is the name used in the `NETNAME()` parameter of your connection definition in your `'SDBxIN00'` (Shadow Direct) or `'SWSxIN00'` (Shadow OS/390 Web Server) exec.

- **MRO Bind Time Security**

  This check will ensure that your Shadow Direct/Shadow OS/390 Web Server is authorized to connect to a CICS system, and is performed at `OPEN_ PIPE` time. If you start your Shadow Direct/Shadow OS/390 Web Server started task, and the `OPEN_ PIPE` request fails with `RESPONSE(SYSTEM_ERROR) REASON(IRC_CONNECT_FAILURE)` and `subreason1= 176 (decimal)`, you will need to set up MRO Bind time security for Shadow Direct/Shadow OS/390 Web Server. The Shadow Direct/Shadow OS/390 Web Server 's userid must have READ authority to profile DFHAPPL.applid, where applid is the id for the CICS region specified in the `APPLID()` parameter of your connection definition in your `'SDBxIN00'` (Shadow Direct) or `'SWSxIN00'` (Shadow OS/390 Web Server) exec.

- **Link Security**

  The target CICS system checks link security against requests from the Shadow Direct/Shadow OS/390 Web Server using the Shadow Direct/Shadow OS/390 Web Server region's userid. These security checks cover the following:

- Transaction attach security (when attaching the mirror transaction)
- Resource and command security within the server application program

If you have transaction and program security enabled in your CICS region, make sure that the Shadow Direct/Shadow OS/390 Web Server's userid can access the mirror transaction and the CICS server program. Also, if the CICS server program accesses any protected resource, make sure the Shadow Direct/Shadow OS/390 Web Server's userid can access the same protected resource.

- **USER Security**

  User security applies only when `ATTACHSEC(IDENTIFY)` is specified on the CICS EXCI connection definition. The userid, which is the same userid used for logging on to the MVS system where your Shadow Direct/Shadow OS/390 Web Server resides, will be passed in the DPL request and will be used to perform the same checks as LINK security.

> ***Note:***
> USER security can never attain more privileges than LINK security allows. It may attain equal or less privilege, but never more.

For more information on EXCI security, refer to the CICS/ESA External CICS Interface Manual.

Verify Installation

For Shadow Server Users:

1. Restart Shadow Server.

2. To verify the connection between the Transaction Server for CICS and CICS, be sure you receive the following messages:

   - In the syslog:
     ```
     for EXCI:
     SDW0358I EXCI SUPPORT ACTIVATED
     ```

   - In the Shadow Direct trace:

```
for EXCI:
INIT_USER EXECUTED - EXCI - EXCI INIT_USER COMPLETED NORMALLY
INIT_USER EXECUTED - EXCI - EXCI INIT_USER COMPLETED NORMALLY
ALLOCATE_PIPE EXECUTED - EXCI A06CICS1 - EXCI ALLOCATE PIPE COMP
OPEN_PIPE EXECUTED - EXCI A06CICS1 - EXCI OPEN PIPE COMPLETED NO
CLOSE_PIPE EXECUTED - EXCI A06CICS1 - EXCI CLOSE PIPE COMPLETED
DEALLOCATE_PIPE EXECUTED - EXCI A06CICS1 - EXCI DEALLOCATE PIPE
```

3. Run the Demo using *Shadow Direct.*

   a. Configure an ODBC data source. For further information, see the Shadow Direct User’s Guide chapter, “Configuring Data Sources”.

   b. Do one of the following, depending on the operating system of the client PC:

   – **Windows 95/NT.** From the client PC Start menu, select **Start**>**Programs**>**Shadow Direct**>**VB Demo Application**. The VB Demo window opens.

   – **Windows 3.x.** Select the Shadow group icon, then double-click the VB Demo application icon. The VB Demo window opens.

   c. In the VB Demo window, select **Connections**>**Add New Connection**. The Select Data Source window opens.

   d. Select a data source using either the File Data Source or Machine Data Source tab, then click OK.

   e. In the input box (the white area directly below the Current Connection listing), type the following:

```
CALL SHADOW_CICS('EXCI','EXCS','EXCI','DFH$AXCS',2,'FILEA
',+'     1','',100,'','DATA')
```

Note:
In the above statement, `FILEA` is followed by three spaces (for a total of 8 characters) and `1` is preceded by five spaces (for a total of 6 characters).

   f. Click **Query**. The data from the sample VSAM file appears in the Query Result window.

4. **For Custom RPC Only:** Run the RPC demo program, `'SDCOCIEC'`, by compiling the COBOL member, located in the `'NEON.SAMP'` file. It is preferable to use LE370 COBOL for MVS, however if this is not available, use COBOL II.

For Shadow OS/390 Web Server Users:

1. Compile `'SWCOCIEC'` from the `'NEON.SAMP'` file. It is preferable to use LE370 COBOL for MVS, however if this is not available, use COBOL II.

2. Using a browser from the client, enter your URL to display the Shadow OS/390 Web Server home page.

3. On the home page, click “Shadow OS/390 Web Server sample applications.”

4. In the sample application, click option 4, “Sample Programs that Access CICS, DB2, and IMS.”

5. Click option 5 to execute `’SWCOCIEC’` and view the VSAM data from CICS.

SHADOW_CICS RPC

Parameter	Definition
‘NNNN’	Connection-type as defined in the SD exec, “EXCI”.
‘CCCC’	Connection-name as defined in the SD exec.
‘TTTT’	Tran-ID as defined in CICS.
‘PPPPPPPP’	Program name as defined in CICS.
‘1’	First parameter expected by the program.
‘2’	Second parameter expected by the program.
‘3’	Third parameter expected by the program.
‘4’	Fourth parameter expected by the program.
‘5’	Optional, indicates the length of the commarea. If not present the default is 32k.
‘6’	Indicates if recursive execution of the transaction is required. Possible value is “Y” for yes and “N” for no. The default value is no.
‘DATA’ or ‘MAP’	Column name or map name to be used for the returned data. For the `MAP` keyword, the syntax is: `’MAP(NAME(EXCI) FIELDS(*))’` **NAME.** This entry should correspond to the name assigned to the map during extraction. **FIELDS.** There are two ways to return data from all columns that are enabled in the map definition: • Use an asterisk after FIELDS. • Leave out FIELDS altogether. To exclude some columns, enter the names of the enabled columns you want returned in the parentheses after FIELDS. For more information about the Data Mapping facility, refer to the appropriate chapter in the *Shadow Server User’s Guide*.

Example of EXCI demo transaction:

```
CALL SHADOW_CICS(’EXCI’,’EXCS’,’EXCI’,’DFH$AXCS’,2,’FILEA   ’,
+’     1’,’’,100,’’,’DATA’)
```

Note:
In the above statement, `FILEA` is followed by three spaces (for a total of 8 characters) and `1` is preceded by five spaces (for a total of 6 characters).

Load Balancing

Transaction Server for CICS support includes the ability to direct CICS requests to multiple address spaces. To accomplish this, a Group Director has been added to this component. This Group Director, which is a user-defined server, can pass connections to the best candidate server in the group. It also gives the user the option to not run any application work in a server that accepts inbound connections.

EXCI Failover

The EXCI Failover enables Web/ODBC CICS applications to deliver unparalleled redundancy, recovery, and integrity. To accomplish this, an instance of Shadow Server or Shadow OS/390 Web Server defines an alternate CICS for each CICS connection. If access to a primary CICS connection fails, a hot failover is performed to the alternate CICS, which has been indentified during the installation process. This request is then processed on the alternate CICS connection, and the user is unaware of the failed attempt of the primary CICS connection. Any pipes to CICS which are not serving an in-flight transaction are switched over to the alternate CICS by a built-in EXCI monitor, which is part of the CICS support.

APPENDIX F: *Configuring Secure Sockets Layer (SSL) Support*

To configure the Shadow Server for Secure Socket Layer (SSL) support, you should first complete installation of the server and verify that it can be started and stopped successfully. The following steps are performed, after initial product installation, to configure SSL support for the server.

- **Step 1:** Make Language Environment for MVS & VM run-time libraries available.
- **Step 2:** Merge the SSL load modules into the product’s load library.
- **Step 3:** Set up proxy userid for access to the protected files.
- **Step 4:** Execute SSL utilities:
  - Create a private encryption key.
  - Create a Site Certificate Request.
  - Apply for, and obtain a Signed Site Certificate from a third-party Certificate Signing Authority such as Verisign or Thawte.
  - Generate a Self-signed Certificate for use on your corporate Intranet.
- **Step 5:** Modify the server start-up JCL to allocate the Private Encryption Key and Server Certificate datasets to the server.
- **Step 6:** Change server start-up parameters to enable and configure SSL support.

Step 1: Make available language environment for MVS & VM run-time libraries.

The SSL support routines are written in C, and have been compiled to execute with the Language Environment For MVS & VM, Version 1.5, run-time libraries. Ensure that these libraries are installed on your system.

Normally, these libraries will already be within the `‘LNKLST’` concatenation on your MVS system. If they are not present in `‘LNKLST’` or `‘LPA’`, place the L/E run-time load library into the Shadow Server’s STEPLIB concatenation.

Step 2: Merge SSL load modules into product's load library.

Because of U.S. Government export regulations, the executable load modules for SSL support are distributed on a separate tape. The SSL executable modules must be copied from the distribution tape into the same APF-authorized load library in which other product load modules reside.

Note:
The SSL modules must reside in the same PDS load library in which the remaining product load modules are installed. If you attempt to use a separate load library, the ISPF interface of the product will not operate correctly.

The SSL distribution tape contains a single IEBCOPY format load library. The load modules within this library must be merged into the server's run-time load library.

The `'INSTASSL'` member of the CNTL distribution library contains sample JCL which can be tailored to merge these modules from the distribution tape, into the existing product load library. You must tailor this sample JCL before submitting it for execution.

- Change the `TAPEVOL=` parameter to specify the volume serial number of the SSL distribution tape. The volume serial number is on the external tape label.
- Change the `TAPEUNT=` parameter to specify the input tape unit name.
- Change the `DISKPFX=` parameter to specify the same disk prefix name used in the initial installation job stream.
- Change the `DISKUNT=` parameter to specify the unit name for disk datasets.

Step 3: Set up proxy userid for access to the protected files.

During operation, the Shadow Server will need to open and read a dataset which contains the SSL Private Encryption Key. Access to this dataset should normally be tightly controlled, available only on a need-to-know basis. Exposure of the private key could allow unauthorized parties to impersonate the server or decrypt sensitive message traffic.

The userid associated with the Shadow Server address space should not be authorized to open this dataset. For Shadow OS/390 Web Server, the "default RUNAUTH userid" must also be prohibited from accessing this file.

NEON recommends that you create an MVS Security Subsystem (like, RACF, ACF/2, or TopSecret) userid which is used solely to read this dataset during server start-up processing. Configure the server to use this special userid as a proxy when this dataset is opened.

Step 4: Execute SSL utilities.

Note:

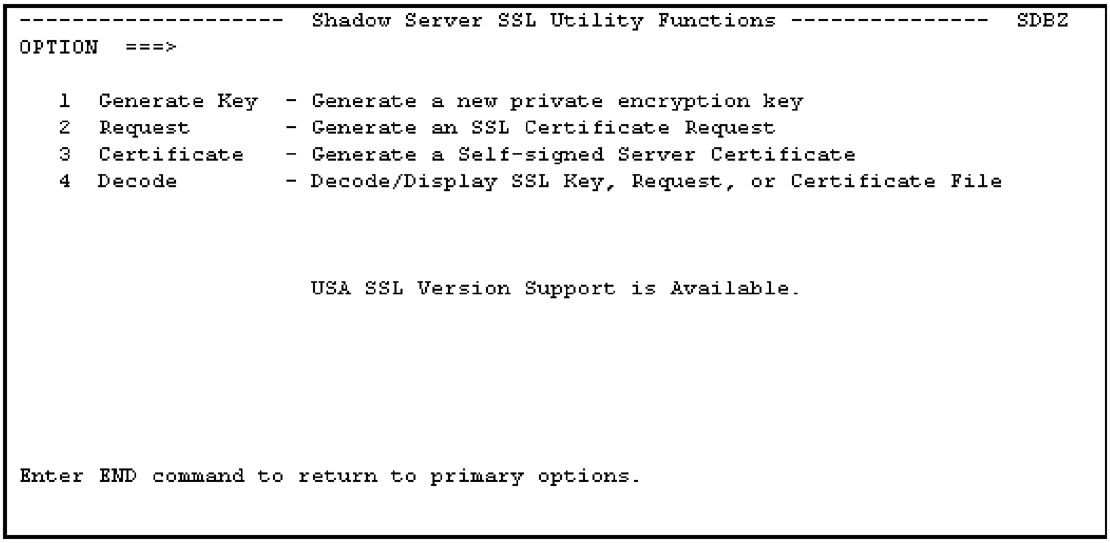
Normally, the following utility functions should be executed by your MVS Security Administration staff, not by the System's Programmer. Access to the Private Encryption Key file should be strictly controlled at all times.

A full discussion of private key cryptography and the secure sockets layer (SSL) protocol is beyond the scope of this installation procedure. This section assumes you have a basic understanding of how Site Certificates are used to verify connections and encrypt message traffic. Information about public key cryptography, server-related security issues, and the SSL protocol abounds on the Internet. You may wish to familiarize yourself with the basic concepts before proceeding.

For SSL support to be operational, the Server needs to do the following:

- Generate a private encryption key.
- Generate a Site Certificate Request file.
- Obtain a signed Site Certificate from a third-party Certificate Signing Authority.
- Generate a self-signed Site Certificate.

These options are found in the Shadow Server Control application (ISPF function 5.14, SSL Utilities), as shown in Figure F–1.



```
--------------------  Shadow Server SSL Utility Functions ---------------  SDBZ
OPTION  ===>

   1  Generate Key  - Generate a new private encryption key
   2  Request       - Generate an SSL Certificate Request
   3  Certificate   - Generate a Self-signed Server Certificate
   4  Decode        - Decode/Display SSL Key, Request, or Certificate File



                      USA SSL Version Support is Available.






Enter END command to return to primary options.
```

Figure F–1. SSL Utility Functions

The Private Encryption Key is a secret binary value used to encrypt message traffic. It must be kept secure at all times from unauthorized access.

The Site Certificate contains public information which allows a client to verify that the partner in a communications session is not being impersonated by an unauthorized site or person. The Site Certificate contains:

- Shadow Server‘s registered Internet domain name.
- Additional site-related identification information.
- The Public Encryption Key - the counterpart to the Private Key.
- Information about the *signing authority* which issued the site certificate.

Shadow Server’s ISPF interface provides four utility functions which you can use to create or obtain a Private Encryption Key and Server Site Certificate. Each utility function is described in the sections which follow.

Most of the input fields on the ISPF panels are self-explanatory. Unless there is some special reason not to do so, we recommend that you store each of these items in its own, separate MVS dataset and use the default RECFM, LRECL, and BLKSIZE values.

Note:

If you are using an OEM Certificate Server to generate a private key or a site certificate, you do not need to execute the SSL utility functions. Make the private key and site certificate available as described in “Step 5: Modify Server start-up JCL.” on page F-14.

The section, “Using OEM Key and Certificate Files” on page F-15 explains how to ensure that OEM files are uploaded to MVS in a format acceptable to Shadow Server.

Generate a Private Key

The first step to perform is generation of a Private Encryption Key. **Once generated, this key must be kept secure, and must not be lost.**

- If the key contents is compromised, a Site Certificate generated from it can no longer be considered secure and must be replaced.

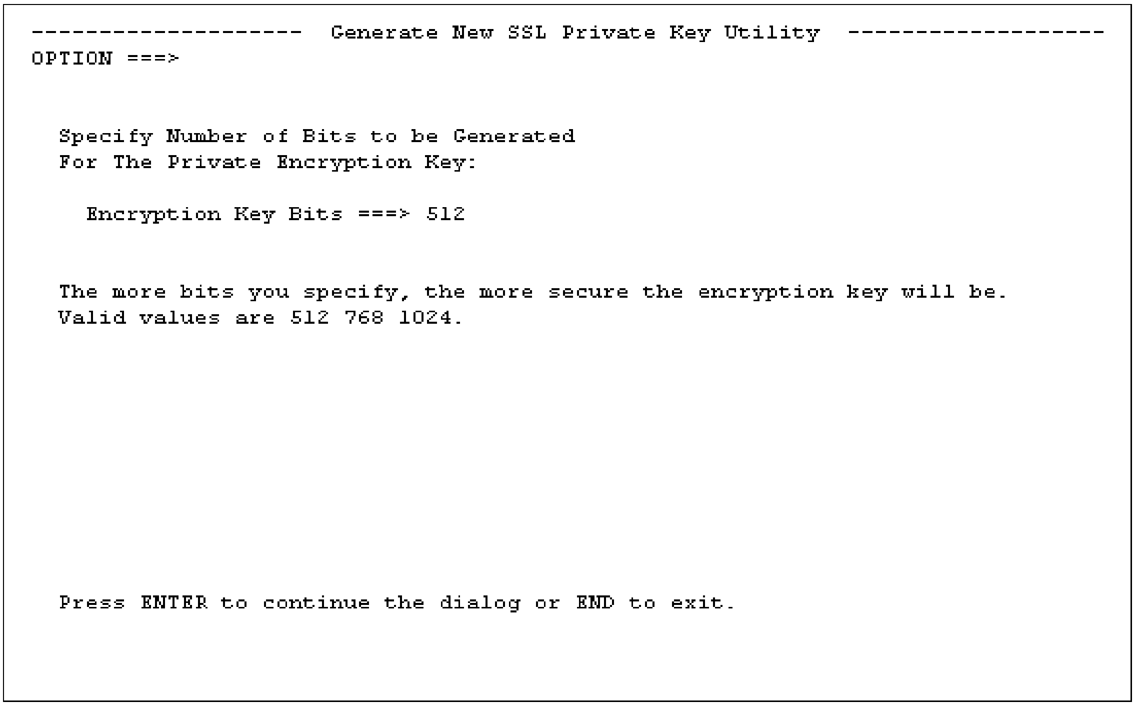
- If it is lost, a Site Certificate generated from it will be useless. The public encryption key within the Site Certificate must be the mathematical counterpart to the private key. Data stream encryption and decryption cannot be performed unless the public key is the mathematical counterpart of the private key.

To generate a private encryption key:

1. Use the ISPF 5.14.1 function to generate a new private encryption key. When specifying the encryption key size (see Figure F–2), you may wish to consider the following:

   - If you use a third-party signing authority (such as Verisign, Thawte, etc.), you should specify the highest allowed key size. The maximum size for domestic SSL support is 1024 bits; The maximum size allowed for non-USA SSL support is 512 bits, since U.S. law prohibits distribution of software supporting larger key sizes.

   - If you are configuring SSL support *only* to encrypt data traffic on your corporate Intr**a**net, *and only* with a self-signed certificate, consider using the smallest allowed key size (512). Smaller key sizes require less CPU time when each SSL connection is established. Note, however, that smaller key sizes pose a greater risk that the encrypted traffic can be decrypted by an un-authorized third-party.

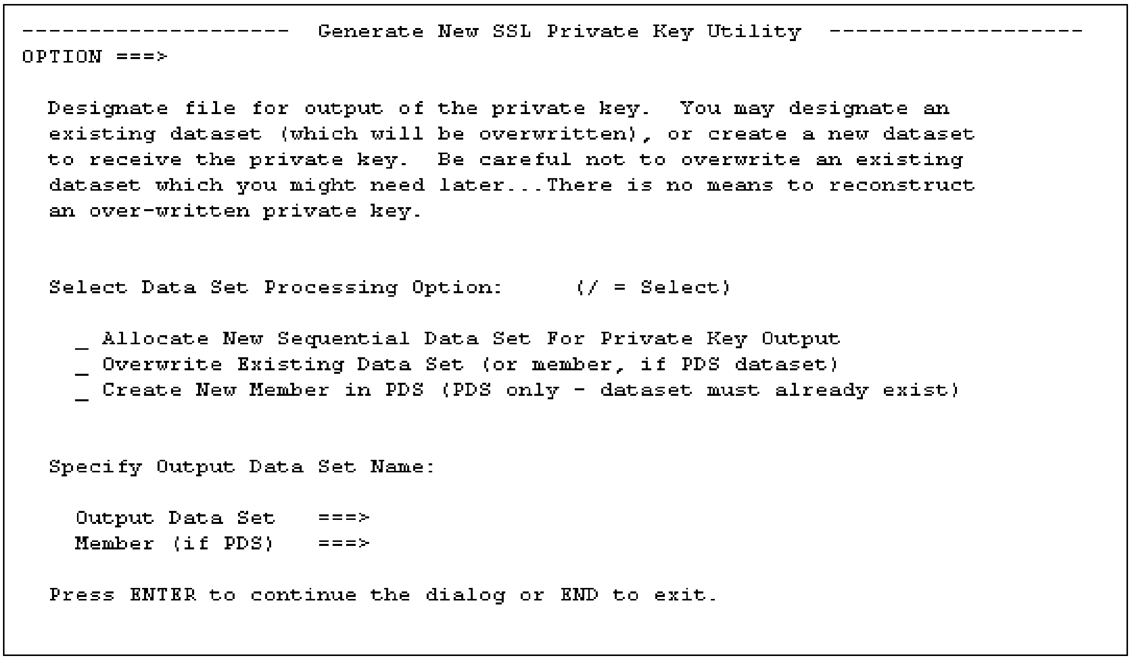


```
--------------------  Generate New SSL Private Key Utility  -------------------
OPTION ===>


  Specify Number of Bits to be Generated
  For The Private Encryption Key:

    Encryption Key Bits ===> 512


  The more bits you specify, the more secure the encryption key will be.
  Valid values are 512 768 1024.









  Press ENTER to continue the dialog or END to exit.
```

Figure F–2. Specifying Size of Encryption Key

2. After you specify the key size, press <ENTER> to take you to the next panel (see Figure F–3) where you will:

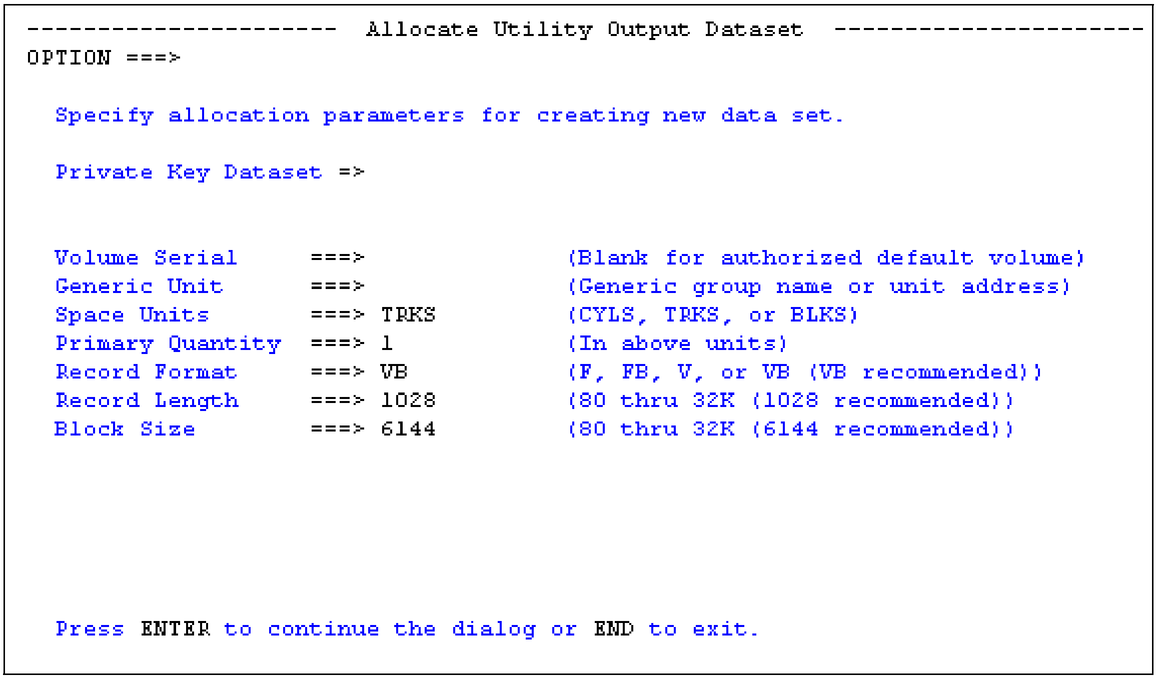
   - Select a data set processing option.
   - Specify an output data set name.



```
--------------------  Generate New SSL Private Key Utility  -------------------
OPTION ===>

  Designate file for output of the private key.  You may designate an
  existing dataset (which will be overwritten), or create a new dataset
  to receive the private key.  Be careful not to overwrite an existing
  dataset which you might need later...There is no means to reconstruct
  an over-written private key.


  Select Data Set Processing Option:     (/ = Select)

    _ Allocate New Sequential Data Set For Private Key Output
    _ Overwrite Existing Data Set (or member, if PDS dataset)
    _ Create New Member in PDS (PDS only - dataset must already exist)


  Specify Output Data Set Name:

    Output Data Set    ===>
    Member (if PDS)    ===>

  Press ENTER to continue the dialog or END to exit.
```

Figure F–3. Specifying Data Set Information

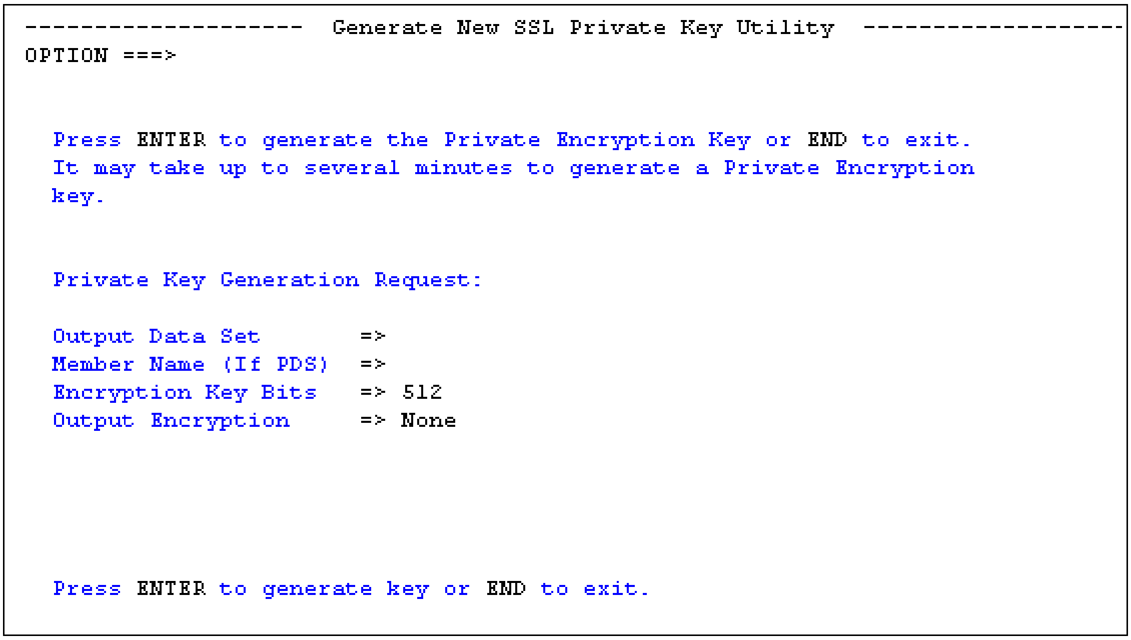
3. Press <ENTER> to continue. On the next panel, you will specify allocation parameters for creating the new data set.



```
----------------------  Allocate Utility Output Dataset  ----------------------
OPTION ===>

  Specify allocation parameters for creating new data set.

  Private Key Dataset =>


  Volume Serial      ===>              (Blank for authorized default volume)
  Generic Unit       ===>              (Generic group name or unit address)
  Space Units        ===> TRKS         (CYLS, TRKS, or BLKS)
  Primary Quantity   ===> 1            (In above units)
  Record Format      ===> VB           (F, FB, V, or VB (VB recommended))
  Record Length      ===> 1028         (80 thru 32K (1028 recommended))
  Block Size         ===> 6144         (80 thru 32K (6144 recommended))






  Press ENTER to continue the dialog or END to exit.
```

Figure F–4. SSL Allocation Parameters

4. Press <ENTER> to access the next panel, which shows you the private key generation request.



```
--------------------  Generate New SSL Private Key Utility  ------------------
OPTION ===>


  Press ENTER to generate the Private Encryption Key or END to exit.
  It may take up to several minutes to generate a Private Encryption
  key.


  Private Key Generation Request:

  Output Data Set        =>
  Member Name (If PDS)   =>
  Encryption Key Bits    => 512
  Output Encryption      => None




  Press ENTER to generate key or END to exit.
```

Figure F–5. Private Key Generation Request

5. Press <ENTER> to generate the key. It may take up to several minutes to do this. You will receive a message telling you that the key has been successfully generated and the location where it has been stored.

Once generated, the Private Encryption Key will look like this:

```
-----BEGIN RSA PRIVATE KEY-----
MIIBOwIBAAJBAMFfjhBwuLTEpAYbE2/ZNp/Y1FeQOdVKiEC/QLA+SeZ9C8P+x7Je
WpmHHwtxPzaMN05cAFxJaSkeDuMaXvh/PdMCAwEAAQJAYgNO/KJF3Mo24SlkJrMQ
buD2cBOYXLXUbg0CetJ3nq0tDTlx2iP/QZq938nIyrJgc3pjuBH1zaRNAgsvseIB
8QIhAOxdaekpzTglvnvELip3VBMBucUPhcvpLV3LPPOUXNmHAiEA0W/fNwJZS/ak
lw0v7udM5QtOOEKIOkip0PcgIWIt3FUCIQDgqkXiNzZ2OQIeuDE9ciD60/gUxBVj
5aaWcXjk+c9rfwIhAMY8bbtTrdIJEqEnCkfHBzGFEfSOhQMl5Ba20uLGW0JBAiBT
bBhCbOF914bri0x5zIdSFpdEXFPHIl/CEXwYKtVpYA==
-----END RSA PRIVATE KEY
```

The Private Encryption key should be protected using services of your MVS security subsystem (RACF, ACF/2^{TM}, TopSecret^{TM}, etc.).

- READ-ONLY access to the private key should be granted only to security administration personnel who require it.
- The Server software can be configured to access this file under the authority of a proxy userid. *(We strongly recommend that the proxy mechanism be used!)* If a proxy can be used, grant the proxy READ-ONLY access to the private key.

- Do **not** grant **any** access of this file to either the Shadow Server started-task userid or the Shadow OS/390 Web Server “default RUNAUTH userid.”

Generate a Site Certificate Request file.

The second step is generation of a Site Certificate Request file. The Site Certificate can be a third-party signed certificate (ISPF function 5.14.2 of the Primary Options Menu) or a self-signed site certificate (ISPF function 5.14.3 of the Primary Options Menu).

The Site Certificate Request contains encoded information about your server site, such as:

- The Site’s registered Internet domain name.
- The Public Encryption key which is the mathematical counterpart to the Private key.
- Additional site-related identification information.

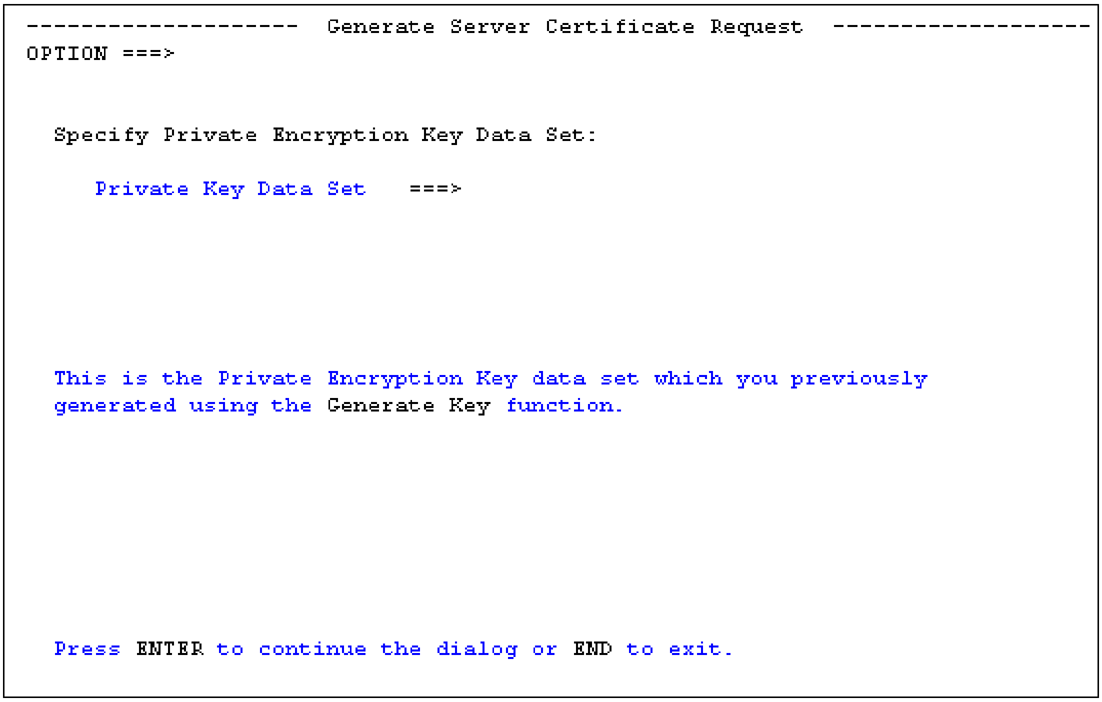
The site certificate signed by a third party commercial authority will insure that no one else impersonates your Server on the global Internet.

You can use the self-signed site certificate while you are applying for a third-party certificate, or you can use it within your own corporate Intranet for the purpose of encrypting sensitive internal data traffic. A self-signed certificate can generally be used only on an Intr**a**net, since no guarantee of trust has been asserted by a widely known third-party. Also, since HTML browsers will not be pre-configured with the signing authority’s (your own organization) identity, they will not automatically establish a communications session with the Shadow Server.

To Obtain a third party signed Site Certificate:

1. Use the ISPF 5.14.2 function to request a third-party Server Certificate. The first panel you will see contains information about the certificate request. Before you can generate the certificate request, you must have already generated a private encryption key.

2. Press <ENTER> to continue to the next panel, where you will enter the private encryption key data set.



```
--------------------  Generate Server Certificate Request  --------------------
OPTION ===>


  Specify Private Encryption Key Data Set:

     Private Key Data Set   ===>







  This is the Private Encryption Key data set which you previously
  generated using the Generate Key function.









  Press ENTER to continue the dialog or END to exit.
```

Figure F–6. Private Encryption Key Data Set

3. After you specify the private key data set, press <ENTER> to take you to the next panel where you will:

   - Select a data set processing option.
   - Specify an output data set name

   This panel contains the same information as the panel shown in Figure F–3.

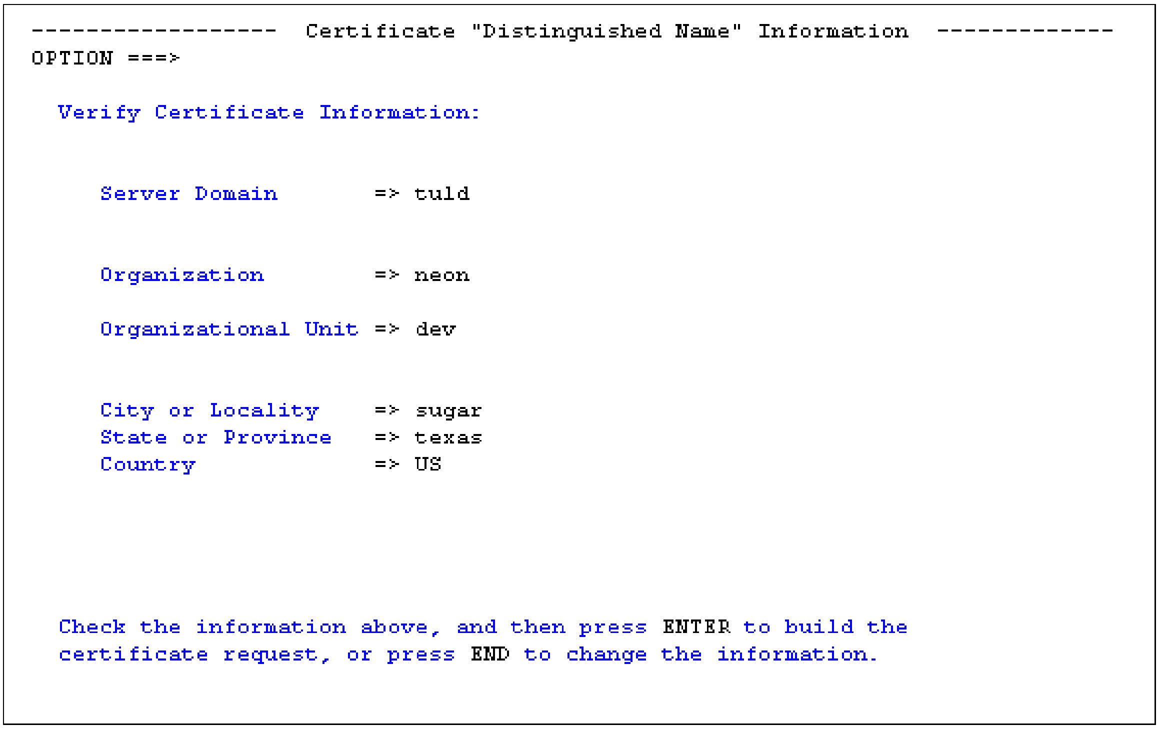
4. Press <ENTER> to continue to the next panel, where you will specify allocation parameters for creating the new data set. See Figure F–4.

5. Press <ENTER> to continue. A dialog panel appears telling you that you will be entering specific information on the three panels that follow. This information includes:

   - Domain name
   - Organization name
   - Organizational unit name
   - City or Locality
   - State or Province
   - Country Code (2 letters)

   Use the legal name under which your organization is registered. Do not use abbreviations for the organization name, or the city or state. Both cities and states must be completely spelled out. You can abbreviate the country, using the proper 2-letter country code.

6. Press <ENTER>. The next panel shows all of the certificate information you have just entered in Step 5:



```
------------------  Certificate "Distinguished Name" Information  -------------
OPTION ===>

  Verify Certificate Information:


     Server Domain       => tuld


     Organization        => neon

     Organizational Unit => dev


     City or Locality    => sugar
     State or Province   => texas
     Country             => US




  Check the information above, and then press ENTER to build the
  certificate request, or press END to change the information.
```

Figure F–7. Third Party Certificate Information

7. Verify the certificate information and press <ENTER> to build the request, or END to change any of the information. If you chose to build the request, you will receive a message telling you the certificate request has been successfully generated and where it is located.

Once the Certificate Request has been generated, you must send the request to a third-party Certificate Signing Authority. The Signing Authority, after investigating your credentials, then issues your Site Certificate.

Many commercial third-party Signing Authorities are in business around the globe. The two most universally recognized authorities are:

- Verisign, Inc. - http://www.verisign.com
- Thawte Consulting cc. - http://www.thawte.com

We recommend that you refer to these web sites before generating a Site Certificate Request. Each Signing Authority has its own rules regarding the exact contents of the Site Certificate Request. For example, Thawte will sign a Site Certificate containing a generic server domain name (e.g., www.neon.*.com); whereas, Verisign does not sign generic Site Certificates. There may also be restrictions on the use of abbreviations and/or punctuation.

Once generated, your Site Certificate Request will look like the following:

```
-----BEGIN CERTIFICATE REQUEST-----
MIIBRTCB8AIBADCBijELMAkGA1UEBhMCVVMxDjAMBgNVBAgTBVRleGFzMRMwEQYD
VQQHEwpTdWdhciBMYW5kMSIwIAYDVQQKExlORU9OIFN5c3RlbXMgSW5jb3Jwb3Jh
dGVkMRgwFgYDVQQLEw9TV1MgRGV2ZWxvcG1lbnQxGDAWBgNVBAMTD3d3dy5uZW9u
ZGV2LmNvbTBcMA0GCSqGSIb3DQEBAQUAA0sAMEgCQQDBX44QcLi0xKQGGxNv2Taf
2NRXkDnVSohAv0CwPknmfQvD/seyXlqZhx8LcT82jDdOXABcSWkpHg7jGl74fz3T
AgMBAAGgADANBgkqhkiG9w0BAQQFAANBAGhbDHbSv7j/qq+GvZXoHNkMKGJNV+A4
MWyKv3R0y87v/gmKh41f1uOdP1kYTm47XmbMDBCoBgyiN/GbehD23CM=
-----END CERTIFICATE REQUEST-----
```

Follow the instructions provided by the Signing Authority for submitting this request file, along with supporting credentials.

Note:

The Certificate Request file will be stored in EBCDIC. You can successfully cut-and-paste this information into an e-mail message or HTML form from a TN3270 session; translation from EBCDIC to ASCII will be done automatically. If you file transfer this information, be sure that EBCDIC-to-ASCII translation occurs.

Once you receive your signed certificate from a third-party Signing Authority, be sure it is uploaded to MVS in a format acceptable to the Server. Consult the section “Using OEM Key and Certificate Files” for details.

To obtain a self-signed Site Certificate:

1. Use the ISPF 5.14.3 function to request a self-signed Server Certificate. The first panel you will see contains information about the certificate request. Before you can generate the certificate request, you must have already generated a private encryption key.

2. Press <ENTER> to move to the next panel, where you will enter the private encryption key data set. This panel contains the same information as shown in Figure F–6.

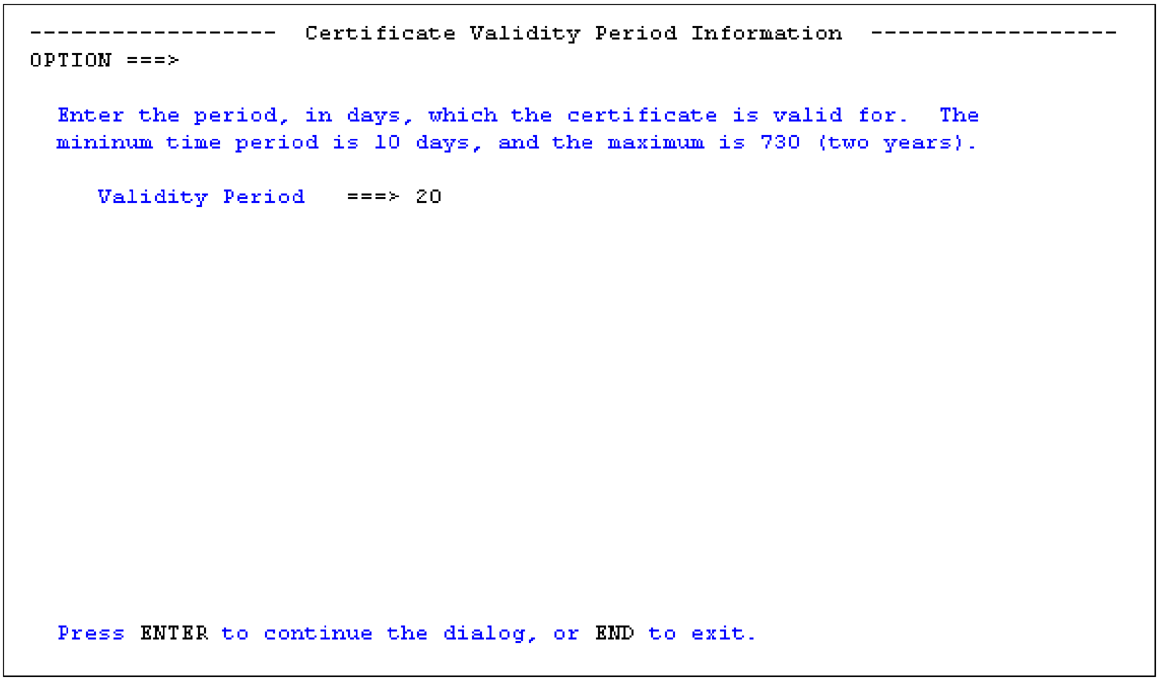
3. After you specify the private key data set, press <ENTER> to take you to the next panel where you will:

   - Select a data set processing option.
   - Specify an output data set name.

   This panel contains the same information as the panel shown in Figure F–3.

4. Press <ENTER> to continue to the next panel, where you will specify allocation parameters for creating the new data set. See Figure F–4.

5. Press <ENTER> to access the next panel. Here you will enter the number of days for which the certificate will be valid.



```
------------------  Certificate Validity Period Information  ------------------
OPTION ===>

  Enter the period, in days, which the certificate is valid for.  The
  mininum time period is 10 days, and the maximum is 730 (two years).

     Validity Period   ===> 20















  Press ENTER to continue the dialog, or END to exit.
```

Figure F–8. Certificate Validity Period

6. Press <ENTER> to continue. A dialog panel appears containing values for the following Data Sets:

   - Key Data Set
   - Request Data Set
   - Output Data Set

7. After verifying the datasets, press <ENTER> to build the request or END to change any of the information. If you chose to build the request, you will receive a message telling you the certificate request has been successfully generated and where it is located.

Once generated, the Site Certificate will look like the following:

```
-----BEGIN CERTIFICATE-----
MIICADCCAaoCAQAwDQYJKoZIhvcNAQEEBQAwgYoxCzAJBgNVBAYTAlVTMQ4wDAYD
VQQIEwVUZXhhczETMBEGA1UEBxMKU3VnYXIgTGFuZDEiMCAGA1UEChMZTkVPTiBT
eXN0ZW1zIEluY29ycG9yYXRlZDEYMBYGA1UECxMPU1dTIERldmVsb3BtZW50MRgw
FgYDVQQDEw93d3cubmVvbmRldi5jb20wHhcNOTcwOTA0MjEwNjA1WhcNOTgwOTA0
MjEwNjA1WjCBijELMAkGA1UEBhMCVVMxDjAMBgNVBAgTBVRleGFzMRMwEQYDVQQH
EwpTdWdhciBMYW5kMSIwIAYDVQQKExlORU9OIFN5c3RlbXMgSW5jb3Jwb3JhdGVk
MRgwFgYDVQQLEw9TV1MgRGV2ZWxvcG1lbnQxGDAWBgNVBAMTD3d3dy5uZW9uZGV2
LmNvbTBcMA0GCSqGSIb3DQEBAQUAA0sAMEgCQQDBX44QcLi0xKQGGxNv2Taf2NRX
kDnVSohAv0CwPknmfQvD/seyXlqZhx8LcT82jDdOXABcSWkpHg7jGl74fz3TAgMB
AAEwDQYJKoZIhvcNAQEEBQADQQCQqDZhuXdiuWq5r9RNDwR2AqhMtYRrQfAy53Yh
pqUYNXaq40TG2jmgabW1/6HiJvReTKCebPOWPSRjcWelIUd9
-----END CERTIFICATE-----
```

When using a Self-Signed Certificate in conjunction with Shadow OS/390 Web Server:

- If a Netscape Browser is being used to contact the server, the browser will prompt you to either accept or reject the connection request. Since the browser will not know the identity of the signing party, the end user is given the option of accepting or rejecting the connection.
- If a pre-version 4.x Microsoft browser is being used, the browser will unconditionally reject connection with the server. You must first *prime* the browser with the identity of the signing authority before you can establish an HTTP connection. This procedure is described later on in this chapter.
- After you create your self-signed site certificate, we strongly recommend that you use a recent version of Netscape or Microsoft Internet Explorer browser for the initial testing. Confirm proper operation of SSL and the site certificate before attempting to use a pre-version 4.x of Microsoft Internet Explorer. This allows you to gain a feel for how browsers, in general, react to an unknown certificate signing authority. It also allows you to defer generation and transmission of a DER-encoded certificate until you are certain that the certificate has been generated successfully and the server is properly configured for SSL support.

Create DER-encoded Certificate Copy

You will only need to execute this function when using a self-signed Site Certificate in conjunction with earlier versions of the Microsoft Internet Explorer browser.

As mentioned in the previous section, Version 2.x Microsoft browsers unconditionally reject SSL connection establishment to sites where the certificate signing authority is unknown. Before you can establish an SSL connection to your Server, you must first make the certificate signing authority known to these versions -- I/E.

This is done by transferring the binary DER-Encoded certificate to the browser, using HTTP, as MIME type “application/x-x509-ca-cert”.

Execute the DER-encoding SSL utility to obtain the converted Site Certificate data. For Shadow OS/390 Web Server, see the supplied SELFSIGN member in the WWW master rule set for information on how to accomplish the transfer to an I/E web browser.

Step 5: Modify Server start-up JCL.

In order to start the Shadow Server with SSL support configured, you must make the following dataset allocations in the start-up JCL. Keep in mind that `xxx` is the three character product id (SWS for Shadow OS/390 Web Server and SDB for Shadow Direct).

DD Name	Dataset
xxxPKEY	Allocate the Private Encryption Key used to generate the Site Certificate Request and/or create the Self-Signed Certificate. This dataset should be set up as `DSORG=PS,RECFM=VB,LRECL=80,BLKSIZE=nnn`
xxxCERT	Allocate the Site Certificate which you created by the Self-Sign utility or which you obtained from a commercial signing authority. This dataset should be set up as `DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=nnn`

If you obtain a commercially signed Site Certificate, be certain you do NOT delete the header and trailer lines. These lines, although they appear to be merely commentary in nature, are a part of the certificate.

Note:

The first line should consist entirely of the characters “`-----BEGIN CERTIFICATE-----`”; The last line should consist entirely of the characters “`-----END CERTIFICATE-----`”.

The server, itself, need not (we strongly recommend against) have authorization to open these datasets. Use the SSLUSERID parameter to specify a proxy userid (see next section).

Step 6: Change Server start-up parameters to enable and configure SSL support.

Before starting the Shadow Server, you will need to set the following configuration parameters within the initialization REXX procedure, `‘SWSxIN00’`:

Parameter	Value
SSL	Set to YES to enable SSL support within the server
SSLUSERID	Set to the 1-to-8 character proxy userid to be used when the SSL encryption key file is opened. This userid must have **READ** access to the private encryption key and certificate files.
TRACESSLEVENTS	Set to YES to enable tracing of SSL-related events
TRACESSLACCEPTSTATES	Set to YES to enable tracing of SSL protocol exchanges during session establishment.

Using OEM Key and Certificate Files

If you obtained either an SSL private key or site certificate file from a source other than Shadow Server's SSL Utilities, you must ensure that they are stored in MVS datasets in a format which can be accepted by the server. This section describes the file formats acceptable to the server and some potential problems which you might encounter.

- The private key and certificate files must not vary significantly from the examples shown in the preceding section. Both the key and certificate files must be encoded using the EBCDIC (not ASCII) character set.

- The first and last line of each file consists entirely of human-readable text, such as "`-----Begin Certificate-----`". These text lines must be present exactly as shown, above.

- Each "base64" (or PEM) encoded line, except the last, must break after column 64.

- The files in which the key and certificate information is stored must be native MVS `DSORG=PS` type files, or members of a PDS(E). Each line within the file must appear as a single MVS "logical record". The files must not contain special linefeed (ASCII `X'0A`, EBCDIC `X'25'`), newline (EBCDIC `X'15'`) or carriage return (`X'0D'`) characters which are commonly used by non-MVS file systems as these special characters do not support the concept of "logical records" used for native MVS sequential files.

- If the key or certificate file is defined as `RECFM=V` or VB, no trailing blanks can be present at he end of each data line. The logical record length must equal exactly the number of non-blank data bytes in each record. (You can edit a variable-length file using ISPF and issue the "save" command to ensure that trailing blanks are stripped and are not included in the logical record length.)

- The Shadow Server does not support 2nd-Level DES encryption of the private key file; it must be encoded as "base64" (or "PEM") text. This eliminates the need to specify the DES decryption key via an operator console or a potentially insecure configuration file during server start-up. Again, we strongly recommend that you use RACF or other MVS Security Subsystem product to restrict access to the private key file.

Copyright Notices and Patents For SSL Support

RSA Data Securities, Inc. owns patents, valid in the USA, on certain encryption algorithms which are used within the SSL implementation. NEON Systems, Inc., has licensed RSA Data Security, Inc.'s BSAFETM Cryptographic Toolkit.

SSL support in the Shadow Server is based upon, and incorporates substantial portions of the C-language SSL ("SSLeay") implementation written by Eric A. Young. NEON Systems, Inc. claims a copyright only on the modifications which integrate it into the Shadow Server. We gratefully acknowledge his generosity in making this top-notch implementation of SSL publicly available.

The following is Mr. Young's copyright notice reproduced in full:

Copyright (C) 1995-1996 Eric Young (eay@mincom.oz.au) All rights reserved.

This file is part of an SSL implementation written by Eric Young (eay@mincom.oz.au).

The implementation was written so as to conform with Netscape's SSL specification. This library and applications are FREE FOR COMMERCIAL AND NON-COMMERCIAL USE as long as the following conditions are adhered to.

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this code is used in a product, Eric Young should be given attribution as the author of the parts used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

All advertising materials mentioning features or use of this software must display the following acknowledgment:

This product includes software developed by Eric Young (eay@mincom.oz.au)

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG "AS IS"' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

APPENDIX G:
Link-Editing the DSN3@ATH Exit

This appendix describes the process for link-editing the Shadow Server DSN3@ATH to another module.

Introduction

Shadow Server, by default, automatically propagates userids. However, if your site uses 8-character userids, you will need to turn this feature off. This can be done by entering the following statement into your initialization EXEC:

```
“MODIFY PARM NAME(AUTOUSERID) VALUE(NO)”
```

If you turn off Shadow Server’s automatic userid propagation, you will need to depend on a DSN3@ATH exit to propagate your userids.

The Shadow Server exit routine is supplied in load module format and must be link-edited with the existing DSN3@ATH load module. It does not replace your current exit; it merely front-ends it. After Shadow Server's CSECT has completed its work, it branches to your code.

The Shadow Server DSN3@ATH exit is designed to be link-edited with another module. It cannot be installed by itself. If you try to use the product exit by itself, you will cause DB2 to reject all attempts for access.

To link-edit your DSN3@ATH exit with Shadow Server's, use the LINKAUTH JCL provided in the CNTL library. Before you run it, you must modify LINKAUTH in the following way:

1. Change the job card for your data center's standards.
2. Modify the DSN parameter of the `‘DB2LIB’` DD card to the name of your production DB2 load library.
3. Alter the DSN parameter of the `‘SDBLOAD’` DD card to the name of your production Shadow Server load library.
4. Change the DSN parameter of the SYSLMOD DD card to the name of the target DB2 load library.

Note:
The target load library should be a test library until testing is completed.

5. Customize or remove the `CHANGE DSN3@ATH(DB1@ATH)` control card. This control card is used to change the external reference in the Shadow Server SDB2AUEX CSECT that points to the next CSECT to run. Usually, the next CSECT is DSN3@ATH, in which case you must remove the CHANGE card. If on the other hand the next CSECT to run is not DSN3@ATH, you must use the change card to specify the correct CSECT.

Adding the Subsystem ID to SYS1.PARMLIB

Shadow Server runs as a formal MVS subsystem. Formally, such subsystems are supposed to be defined at IPL time with statements in the `'IEFSSNxx'` member of `'SYS1.PARMLIB'`. In practice, many subsystem IDs are added dynamically.

If your particular installation does not allow for the dynamic addition of subsystem IDs, you can do this manually. Just insert the following sample statement into the production `'IEFSSNxx'` member of `'SYS1.PARMLIB'`:

```
SDBB      PRODUCTION Shadow Server
```

Binding a Plan with the Repeatable Read Option

In almost all cases, applications will use a plan bound with the cursor stability option. However, if your application requires a plan bound with the repeatable read option, follow the steps below.

Bind the Plan

While you are in the Bind screen of the DB2I application, fill in the fields as follows:

LIBRARY Give the name of the Shadow Server DBRM library.

MEMBER Specify OPRXSQ (unless you are running DB2 Version 2.2, in which case specify OPRXSQ22).

PLAN NAME
Specify `'SDxR1010'`. Here, the x must be replaced with the last character of the Shadow Server subsystem name. The subsystem name is usually `'SDBB'`; therefore, `x` will probably be “`B`”).

Note:
The fourth letter in this plan name, “`R`” indicates that the cursor stability option is being used.

ISOLATION LEVEL
Specify RR (for repeatable read).

Granting Access to the Plan

You will need to grant access to your plan. You can do this by following these steps.

1. Go to the SPUFI screen. For more information on SPUFI, see *IBM DATABASE 2 Version 2 Application Programming and SQL Guide* (SC26-4377-02).

2. Execute the command:

```
GRANT EXECUTE ON PLAN (SDxR1010) TO PUBLIC
```

Providing an ACF2 Command Limiting List

If you are running ACF2 and are using a command limiting list, you will need to add some entries to allow ISPF/SDF to function.

The commands shown in the table below should be added to your command limiting list.

Command	Description
DSN	The DSN command processor. On systems where DB2 is installed, this command should already be authorized.
SDB	The command for invoking the ISPF/SDF application.
SDB2RU	An internal component of the SDB command.
SDBORU	An internal component of the Trace Browse facility.
SDISCBRU	An internal component of the Control Block Display application.
SDISDBRU	An internal component of the Database Control application of ISPF/SDF.
SDISLIRU	An internal component of the Link Control application of ISPF/SDF.
SDISRMRU	An internal component of the Attached/Remote Users Control application of ISPF/SDF.

A sample ACF2 command limiting list is supplied in assembler source format in the SDA2CMLS member of the product ASM dataset.

APPENDIX H: Load Balancing

This appendix describes load balancing, a feature of Shadow Direct, which allows inbound connections to be automatically directed to copies of Shadow Server with the greatest resources available.

Detailed instructions for installing Load Balancing can be found in Chapter 1 of the Shadow Installation Guide.

Load Balancing

For Shadow Direct only

The load balancing feature enables Shadow Server to do the following:

1. Finds the first server’s address space.
2. Checks that server for sufficient virtual storage. If there is enough storage, it is marked as a candidate.
3. Checks the number of active connections.
4. Repeats steps 1-3 until all the candidates are identified.
5. From the available candidates, Shadow Server picks the one with the least number of active connections.

If a candidate can **not** be found, the connection is rejected with an “`Inadequate host resources`” error message.

Load balancing is transparent to ODBC applications running on the client because each application connects to one copy of the server using one port number. The target copy of the server determines whether it, or another copy actually handles the session. If another copy of the server is a better choice, the session is transferred to that copy.

Note:
The ODBC client can be optionally configured with the port numbers of more than one member of the group to improve reliability by providing a fallback if the copy of the server that uses the base port number is not available.

Benefits of Load Balancing

Load balancing greatly increases the number of concurrent connections Shadow Direct can handle. It allows more connections using RPCs to be concurrently handled. This is a key point, because ODBC connections using RPCs exhaust the virtual storage resources of any one address space MUCH faster than DB2 only ODBC connections.

The Group Concept

Load balancing is based on the concept of a group. All copies of the server on one system with the same group name are automatically members of the same group. A copy of the product can be a member of only one group at a time or it can be configured to be a stand-alone server and not as a member of any group. All address spaces in a group must reside on the same MVS image.

The group name can be changed at any time, which means copies of the server can join groups or leave groups as needed.

Note:

This feature is only supported for servers using IBM TCP/IP or OE Sockets.

Load balancing is not supported for LU 6.2, Interlink TCP/IP, or Network Systems TCP/IP.

Error messages for insufficient virtual storage are always returned to the ODBC client, regardless if the server is a member of a group or not.

Adding Shadow Server to a Group

In order to add a Shadow Server to a group, the following parameter must be set within the Shadow Server initialization exec, `‘SDBxIN00’`:

```
“MODIFY PARM NAME(GROUPNAME)   VALUE(xxxxxxxx)”
```

You can also add or change a server’s group dynamically by using the ISPF panels (**option 5.2**, select **prodparm** and change the **LOAD BALANCING GROUP NAME**).

See *Started Task Parameters* for more information on GROUPNAME.

Controlling How UDP and TCP/IP are Used

The NETMODE parameter controls how UDP and TCP/IP are used. The modes control if the main address space handles UDP or TCP/IP sessions and how many tasks are used to accept inbound sessions. For Load Balancing this parameter should be set to TCPMAIN for IBM IUCV TCP/IP or IBM OE Sockets.

```
"MODIFY PARM NAME(NETMODE)  VALUE(TCPMAIN)"
```

APPENDIX I: WLM Enablement

This appendix describes WLM Enablement, a policy driven manager system resources which allows a user to define system performance goals in the same terms that would be used in a service level agreement.

WLM Enablement for Shadow Servers

The WorkLoad Manager (WLM) is a component of the OS/390 operating system, first introduced in MVS/ESA 5.1. The WLM functions used by the Shadow Servers are only available in ESA 5.2 and above.

> ***Note:***
> Load balancing automatically directs the individual in-bound transaction to the copy of Shadow Server with the greatest resources available. See Appendix H, "Load Balancing," for more information.

WLM operates in one of two modes:

- **Goal mode.** System performance goals are stated in a new entity called a WLM service policy. WLM manages the system resources to meet those goals. SRM specifications in `'SYS1.PARMIB'` members `'IEAIPSxx'` and `'IEAICSxx'` specifications are ignored.
- **Compatibility (usually called compat) mode.** System performance is managed by the SRM (business as usual), based on the parameters in `'SYS1.PARMLIB'` members `'IEAOPTxx'`, `'IEAICSxx'` and `'IEAIPSxx'`. If you activate a WLM service policy in compat mode, a subset of WLM functions become available. This compat mode support is enabled in the Shadow Server implementation.

> ***Note:***
> Our WLM support only works without our address space and does not sysplex enable us.

Without this Support

WLM manages the server space as an entity, and has no awareness of individual transactions. This means:

- For transactions of the same type, response times could vary widely as the server address space handles varying workloads.

- There would be no easy way of controlling access to resources, or accounting for resource usage at the transaction level.

Enclaves

To facilitate implementation of transaction management, WLM provides a set of services for managing work in entities known as enclaves. An enclave is a group of one or more MVS TCBs and SRBs that are logically related, (usually through working on the same logical unit of work).

An enclave can be long or short lived. In the Shadow Server implementation, an enclave exists only for the duration of the time that a transaction is being processed.

Initialization Parameters

The new initialization parameters are covered in Appendix B of the Shadow Server User's Guide. To enable the WLM support, set WLMCONNECT to YES in your initialization parameters. If WLMCONNECT is set to NO, everything is off.

WLM Service Definition

The WLM service definition is the repository for service policies. There can be:

1. Only one service definition per sysplex.
2. Only one policy from that definition can be active in a sysplex at any time.

The service definition is stored in WLM couple datasets. An active service policy is required for WLM support in both goal and compat modes. Two key components of a WLM service policy are its service classes and classification rules.

Service Classes

These are similar to compat mode performance groups. They service class periods where the performance goals are assigned.

Classification Rules

These are used to assign transactions to service classes.

Minimum Definition Requirements

- A classification rule for the appropriate subsystem type (SDB, SWS or other if specified in the Shadow Server's initialization parameters).

- A service class for running Shadow Server transactions. This could be defined specifically for the Shadow Servers, or from an existing one (for example, use a TSO service class and define report classes for separating out the activity at report time).

Note:
IBM recommends setting up no more than 30 service classes.

Samples

The statements are shown as they would appear on the relevant definition screens in the WLM ISPF dialog, provided by IBM for building service policies.

Sample classification rule for Shadow OS/390 Web ServerShadow OS/390 Web Server

This rule assigns all Shadow OS/390 Web Server transactions to service class SWSNORM, and the transactions would be managed to meet the goals of SWSNORM.

```
Subsystem Type . : SWS         Fold qualifier names?   Y  (Y or N)
Description  . . . Shadow direct transactions
Action codes:  A=After    C=Copy         M=Move     I=Insert rule
               B=Before   D=Delete row   R=Repeat   IS=Insert Sub-rule
        -------Qualifier-------------          -------Class-------- Action
Type       Name     Start                          Service     Report
                                         DEFAULTS: SWSNORM     ________
____   1   ___       _______  ___                  ________    ________
```

Sample classification rule for Shadow Direct

This rule assigns all Shadow OS/390 Web Server transactions from any userid beginning with AI38*, to service class SDBHOT, and transactions from all other users to the default service class, SDBNORM.

```
Subsystem Type . : SDB         Fold qualifier names?   Y  (Y or N)
Description  . . . Shadow direct transactions
Action codes:  A=After    C=Copy         M=Move     I=Insert rule
               B=Before   D=Delete row   R=Repeat   IS=Insert Sub-rule
        -------Qualifier-------------          -------Class-------- Action
Type       Name     Start                          Service     Report
                                         DEFAULTS: SDBNORM     ________
 ____  1  UI         AI38*    ___                  SDBHOT      ________
```

Sample Service Class Definition for Shadow Direct

This service class has three periods. The first two have percentile response time goals, while the third is a discretionary goal. This example shows how a service class for high priority SDB transactions could be defined.

```
Service Class Name . . . . . : SDBHOT
 Description  . . . . . . . . . Hot Shadow Direct transactions
 Workload Name  . . . . . . . . ONLINE    (name or ?)
 Base Resource Group  . . . . . ________  (name or ?)
 Specify BASE GOAL information.  Action Codes: I=Insert new period,
 E=Edit period, D=Delete period.
         ---Period---  ---------------------Goal---------------------
 Action  #  Duration   Imp.  Description
   __
   __    1  300         2    90% complete within 00:00:00.500
   __    2  800         4    90% complete within 00:00:02.000
   __    3                   Discretionary
```

Sample Service Class Definition for Shadow OS/390 Web Server

This service class has two periods. The first has a percentile response time goals and the second a discretionary goal. This is an example of how a service class for normal priority Shadow OS/390 Web Server transactions could be defined.

```
Service Class Name . . . . . : SWSNORM
 Description  . . . . . . . . . Normal Web Server transactions
 Workload Name  . . . . . . . . ONLINE    (name or ?)
 Base Resource Group  . . . . . ________  (name or ?)
 Specify BASE GOAL information.  Action Codes: I=Insert new period,
 E=Edit period, D=Delete period.
         ---Period---  ---------------------Goal---------------------
 Action  #  Duration   Imp.  Description
   __
   __    1  300         3    90% complete within 00:00:01.000
   __    2              4    Discretionary
```

PARMLIB Specifications

These are only available in compat mode, and can be used to assign the Shadow Server transactions to a specific performance group. To do this, the new SRVCLASS parameter must be used in the 'IEAICSxx' definition for the shadow subsystem, and it must refer to an appropriate performance group.

Sample IEAICSxx specification for Shadow Direct

```
SUBSYS=SDB
SRVCLASS=SDBNORM,PGN=29
```

Sample EAIPxxS specification for Shadow Direct

```
PGN=29,(DMN=nn,DP=F4,…)
```

In addition to the above, there would have to be an active WLM policy that contained an appropriate classification rule for SDB transactions, assigning them to service class SDBNORM. These specifications would cause all Shadow Direct transactions to be executed in performance group 29, even when the server address space itself may be in another performance group.

Note:
The dispatching priority of the server address space must be greater than or equal to the dispatching priority of the performance group that will be used to execute the transactions.

APPENDIX J:
Setting up Two Shadow Servers

It may be desirable to bring up separate Shadow Servers, either for testing purposes or for distributing your workload. To start up additional Shadow Servers, complete the following installation steps for each additional server you wish to start:

1. Create new separate VSAM datasets, one for each server that is being run (See “Step 5. Create the VSAM Datasets” on page 1-8 of this guide).

2. [*]Bind new DB2 application plans for each server (See “Step 4. Bind Shadow Server Product Plans” on page 1-7 of this guide).

3. Define a new TCP/IP port or VTAM application ID depending on the communication protocol you are using. (See “Step 8. Define TCP/IP Port Number (TCP/IP Only)” on page 1-12 of this guide)

4. Create a new Shadow Initialization EXEC. (See “Step 9. Customize Initialization EXEC” on page 1-12 of this guide).

   The initialization EXEC is a REXX program used to set product parameters. The name of the initialization EXEC must be `‘SDBxIN00’` for *Shadow Direct*, or `‘SWSxIN00’` for Shadow OS/390 Web Server, where x is the last character of the 4 character subsystem ID.

5. Create a new startup JCL procedure. (See “Step 6. Set Up the Started Task JCL” on page 1-9 of this guide).

   The `‘SDBB’` and `‘SWSS’` members of the CNTL library contain sample JCL procedures needed to run the *Shadow Direct Server and Shadow OS/390 Web Server* main address space (started task). The `‘SDBB’` and `‘SWSS’` PROC must be placed in a procedure library that will be searched for by the MVS **START** command (this may be `‘SYS1.PROCLIB’`, but does not have to be). Optionally, you can change the name of the proc to reflect the new server you are starting. You must change the SSID parameter within the startup proc to reflect the additional subsystem name. This name must be `SDBx` or `SWSx` where `x` is alphabetic.

[*] This step is only required if you are using Shadow Servers with two different version numbers or maintenance releases. If the servers are of the same maintenance release, all other datasets may be shared among the different servers, such as the `‘NEON.LOAD’`, `‘NEON.EXEC(FB)’` and `‘NEON.RPCLIB’` libraries.

6. [*]Set up a new Shadow REXX/exec to invoke the ISPF/SDF application (See “Step 10. Set Up the ISPF/SDF Dialogs” on page 1-17 of this guide). Modify the LLIB statement within the REXX/exec to point to the Shadow load library.

[*] This step is only required if you are using Shadow Servers with two different version numbers or maintenance releases. If the servers are of the same maintenance release, all other datasets may be shared among the different servers, such as the NEON.LOAD, NEON.EXEC(FB) and NEON.RPCLIB libraries.

APPENDIX K: *Detailed Description of Shadow Direct Connection Modes*

This appendix presents a detailed description of the connection modes supported by Shadow Direct.

Introduction

Shadow Direct supports a variety of connection modes. These connection modes can be used to solve a wide range of Client/Server problems. The choice of connection mode controls how long each physical host connection (TCP/IP session or LU 6.2 conversation) is maintained and whether SQL operations are combined together. The different connection modes provide many benefits including virtually unlimited stability, Internet/Intranet support, Logical Unit Of Work (LUOW) encapsulation, MQ support, and in many cases improved throughput.

The connection modes are used to solve a number of client/server implementation problems. The choice of connection mode for each application will depend on which problems need to be solved. Of course, each application can use more than one connection mode if need be. The client/server problems and solutions are described in this appendix.

VCF Connection Modes

There are five connection modes. All of these connection modes can be used together with one copy of the Shadow Server concurrently. The connection modes are:

- Permanent
- Block
- Transaction
- Transblock
- Message

PERMANENT

This is the standard default connection mode. In this connection mode, a permanent TCP/IP or LU 6.2 session or conversation is established at the beginning of application execution and the connection is maintained for the life of the application (in other words, until the application terminates the connection). All SQL operations are executed separately with a few (very important) exceptions such as fetching a row. Rows are almost always fetched from local client storage without any network I/O.

The major advantage of the PERMANENT connection mode is that all of the CPU and wall clock time costs of creating a session are incurred only once during application initialization.

One major disadvantage of this connection mode is that all required host resources (network session, DB2 thread, etc.) must be allocated for the life of the application. In addition, update operations are transmitted separately across the network (see the above discussion of locking).

BLOCK

In BLOCK connection mode, a permanent TCP/IP or LU 6.2 session or conversation is established at the beginning of application execution and is retained for the life of the application. This the same behavior as PERMANENT connection mode and has the same advantages and disadvantages. The key difference between BLOCK mode and PERMANENT mode is that, in block mode, SQL operations are either sent individually (SELECTs and CALLs) or grouped together and transmitted at the end of a LUOW.

SQL operations are collected on the client until a group terminating event occurs. The terminating events are any SELECT or CALL, a COMMIT, a ROLLBACK, or turning Auto-Commit ON.

Note:

COMMIT and ROLLBACK are not even passed to the ODBC driver (by the Microsoft driver manager) unless the application is executing in Manual-Commit mode.

Using Block mode can significantly increase performance of repetitive SQL statements, such as inserts and updates. For instance, if the client application will be replicating data from a client application or issuing a large number of insert queries, the normal method would be to send each SQL statement in a separate network roundtrip. With BLOCK mode set, all of these inserts will be blocked up together and sent to the MVS system as one network roundtrip. So, for 500 inserts, you are saving 499 network roundtrips. The determining factor for transmitting the SQL operations is when the client either issues a commit or turns auto-commit ON.

When running in BLOCK mode, the developer needs to ensure that the size of the data being transmitted to the host is no larger than the NETWORKBUFFERSIZE parameter set on the Shadow Server. The default is 40K and can be increased up to 4MB. Care should be taken not to raise this parameter high, since this amount of storage is allocated in MVS virtual memory for each user connection regardless of connection mode. If the size of the data being transmitted exceeds the NETWORKBUFFERSIZE parameter, then Shadow Server server will issue a S0C3 abend at the point the buffer is read.

TRANSACTION

In this connection mode, SQL operations are transmitted individually (same as with the PERMANENT mode). However, the host connection is terminated at the end of each LUOW. This connection mode allows for much greater scalibilty of Shadow Direct. Using this parameter within your application will cause the connection to the server to be broken after every commit operation or by the client application turning auto-commit ON or OFF. If the client application is running in Auto-Commit mode, the connection will be terminated after every SQL statement. Transaction mode solves a number of issues, such as the limit of 2000 DB2 CAF threads, applications leaving locks in DB2, and below-the-line storage limits in Shadow Server.

TRANSBLOCK

TRANSBLOCK uses both the features of TRANSACTION mode and BLOCK mode. In this connection mode, SQL operations are transmitted either individually or in groups (same as BLOCK mode) and the connection is terminated at the end of each LUOW (same as TRANSACTION mode).

MESSAGE

In this mode, all host communication is done using messages. A permanent host connection is never maintained. SQL operations are either transmitted individually or grouped together and sent at the end of a LUOW. When running in message mode only a single network roundtrip is allowed between the client and server. This means that all the data being transmitted to and read from the server must not exceed the NETWORKBUFFERSIZE parameter that is set in Shadow Server. Message mode should be used when specifying Message type of HTTP or HTTPS.

Improving Performance while using VCF

Since some VCF modes (all except BLOCK) cause connections to be closed and reopened based on different LUOWs, there can be significant overhead caused by clients reconnecting, reauthorizing the userid and password, and reopening a DB2 thread.. Shadow Direct solves this problem by providing server-side reuseable connections.

Note:

These parameters can be used whether running in VCF mode or not.

In order to enable this support, the REUSETHREADS parameter needs to be set to YES in the Shadow Server init exec:

```
"MODIFY PARM NAME(REUSETHREADS) VALUE(YES)"
```

With this parameter set to YES, Shadow will maintain the users’ entire connection in a wait state after they disconnect. This includes leaving the DB2 thread open if one was opened for this connection. If the same user reconnects again then the entire connection, including the DB2 thread, will be reused. If another user connects, the connection can still be reused, but in this case the id will go through security checking again and the DB2 thread will be closed and reopened using the authorization of the new user.

If the connection is not reused before the THREADTIMEOUT period, the TCB for that client connection, including DB2 Thread, is closed. THREADTIMEOUT is set on the server via the following statement in the Shadow init exec:

```
"MODIFY PARM NAME(THREADTIMEOUT) VALUE(xxxx)"
```

where `xxxx` is the number of seconds. The default is 300 seconds.

Since connections are used over and over again with REUSETHREADS set to YES, there is always the chance of storage creep within the connection’s TCB. For this reason, the TCB for the connection is closed after reaching the THREADRESUSELIMIT parameter. This parameter defaults to 100 and can be set in the Shadow init exec via the following statement:

```
“MODIFY PARM NAME(THREADREUSELIMIT) VALUE(xxxx)”
```

where `xxxx` is a value from 0 to 100000.

Since VCF was developed to mainly control the number of active connections within Shadow Direct, the server can be defined to only allow a certain number of connections to be active at one time. The TARGETTHREADCOUNT parameter controls how many active connections can be active at one time within Shadow Server. This parameter is considered only if REUSETHREADS is set to YES. If the TARGETTHREADCOUNT parameter is reached, future connections will queue and wait for a connection to become available rather than a hard fail using the DB2CONCURRENTMX parameter. TARGETTHREADCOUNT is set via the following statement in the Shadow init exec:

```
“MODIFY PARM NAME(TARGETTHREADCOUNT) VALUE(xxxx)”
```

where `xxxx` is a value between 1 and 1000. The default is 100.

Server Parameter Considerations

If a VCF application will be making calls to either NEON or DB2 stored procedures, you will need to set the AUTOCOMMITCALL parameter to NO in the Shadow Server Init exec:

```
"MODIFY PARM NAME(AUTOCOMMITCALL)  VALUE(NO)"
```

Failure to do so can cause problems for applications fetching large amounts of data from stored procedures. By default, when running in Auto-Commit mode, a commit is automatically issued after the stored procedure is executed, followed by another commit at Close Cursor. The commit issued after the stored procedure call will cause the connection to be broken after the first buffer of data is sent to the client. Once this data is processed by the client and it attempts to issue another fetch, a cursor error will occur since the original connection was closed. The drawback of not setting AUTOCOMMITCALL to NO is that locks will be held on the data accessed by the stored procedure for the time it takes to fetch the data.

Since running in VCF mode causes connections to be closed between different LUOWs, you will see more SMF records written if you have the SMFNUMBER parameter set in the Shadow init exec. Since Shadow cuts the SMF record at the end of a session, and since these sessions will be opened and closed more frequently, you may see a large increase in SMF activity, depending on the number of transactions being run.

Technical Discussion on Scalability

Traditional client/system environments create a permanent TCP/IP session or LU 6.2 conversation as soon as the application connects to the host system. This session or conversation is retained for the life of application execution. In addition, a database connection and other resources are also obtained at the start of application execution and are also retained for the life of application execution.

If the number of clients or applications per client is large, then the total number of concurrent sessions can become prohibitively large. There are several reasons for this.

- Each part of the network can impose limits on the number of concurrent sessions.
- Even if explicit limits do not exist, the total memory requirements can exceed the resources available.

The same problems can also arise for the database connection. Many databases (including DB2) impose limits on the number of concurrent connections. In addition, each connection requires a substantial amount of memory that must be allocated for the life of application execution.

Note:

Typically, idle database connections do not use any CPU time. Once again, even if the database does not impose explicit limits on the number of concurrent connections, the total memory requirements can exceed the resources available.

Shadow Direct supports potentially unlimited scalability by providing the Virtual Connection Facility (VCF). The VCF allows an arbitrarily large number of applications to execute concurrently by providing a virtual host connection for each application. Each application “thinks” that it has a permanent host connection for the life of application execution. However, virtual host connection is converted to a real host session/conversation only on an as-needed basis. Real host connections are converted back to virtual host connections as soon as the application completes each set of work.

The overall impact of this approach is that an application uses only network and database resources when it is actually active. Idle applications (perhaps waiting for user input) do not use any network or database resources. Of course, the network and host system must still be able to support the maximum number of concurrently active applications and client systems.

Shadow Direct supports the VCF using standard ODBC application programming interfaces. A host connection is always created and then immediately terminated as part of application initialization. The temporary initial host connection is used for password/userid validation and other aspects of initialization such as exchanging Shadow Direct bind information.

Thereafter the host connection is made and broken based on application execution requirements. Normally the host connection is re-established as soon as the application tries to execute a SQL operation. Of course, if SQL operations are being accumulated on the client for encapsulation into an LUOW, then connection restart is deferred until the LUOW is complete. The host connection is terminated at the end of each LUOW.

Note:

In message mode, a separate host connection is established for each message and “lives” only for one message. A thread pool (described below) is provided on the host for minimizing the cost of creating and deleting large number of connections.

Shadow Direct supports the VCF by providing a thread pooling mechanism on the host. Thread pooling is required because the thread creation process is fairly slow. To be precise, operating system thread creation is actually very fast. However, creating a per-thread security environment (building an ACEE via SAF calls) and establishing the database thread (using `CAF DSNALI OPEN`) is quite time expensive (in CPU time and wall clock time).

Since applications using the VCF may have to make and break host connections quite frequently, the thread creation/deletion overhead could be prohibitive. However, the thread pooling mechanism allows fully initialized threads to be reused rather than created/deleted each time. The thread pool automatically sizes itself to meet client requirements. New threads are created as needed to support clients, and idle threads terminate themselves after a reasonable period of time (installation settable). The net effect, is that the thread pool shrinks and expands as needed in order to handle the workload. Threads are also aged to facilitate resource cleanup at task termination. This approach helps to resolve any possible application storage leaks.

Lock Conflicts/Failures/Elongation

In a traditional client/server environment each update operation is transmitted to the host separately. This means that if several tables are being updated as part of one LUOW, network I/O operations will always intervene between the first database update and the subsequent data modification operations. Even if only one table is being updated, a separate network I/O operation will always be required (unless Auto-Commit is active) between the update and the eventual commit or rollback.

The interleaving of database updates and network I/O operations creates several problems. The first problem is that the subsequent operations may never arrive because of network I/O, client system, or client application failures. At best this will cause hung transactions, throughput degradation, lock conflicts, deadlock failures, etc. At worst it could cause an incomplete database update. The second problem is that even if all of the update operations are completed without error and the commit/rollback is executed, locks must be held for very long periods of time. In other words, lock durations are determined by network I/O speeds rather than memory-to-memory machine execution speeds. Lock elongation will almost always degrade throughput and may cause deadlock conditions.

The difference between network I/O speeds and internal machine speeds is quite large. Network I/O operations normally require tens to hundreds of milliseconds to complete. By contrast, internal database operations normally execute in a few hundred microseconds. The ratio of network I/O times to internal machine times may be as high as 1000 to 1. This means that a traditional client/server application may hold locks for 1000 times as long as a monolithic host application.

Some of these problems (hung transactions holding locks) can be addressed using the Shadow Direct Lock Control Facility (LCF). See the description of the LCF elsewhere in this document.

Shadow Direct addresses this problem by encapsulating complete LUOWs inside a single network I/O message. This approach has several impacts. First, no part of a LUOW is executed until the entire message/LUOW has been received by the server. Second, no network I/O is ever required to complete an LUOW once the LUOW has been started. Overall, LUOW encapsulation guarantees that no part of a sequence of database updates will be started until all of the updates have been received by the host and that no network I/O will be required to complete the LUOW.

Of course, the network I/O message may be larger than a single IP datagram or SNA RU. Shadow Direct prefixes each network I/O message with a length field. The length field is used to verify that an entire message has been received from the client before any processing of the message is started.

Shadow Direct supports LUOW encapsulation using standard ODBC application programming interfaces. Database update operations are accumulated on the client system until the client application indicates that a LUOW has been completed. As soon as the application completes the LUOW a network message is constructed and transmitted to the host for execution. Note that multiple update operations are only cached on the client if Manual-Commit mode is active. If Auto-Commit is active, each update is separately transmitted to the host.

Of course, one or more of the update operations in the LUOW may fail. If a failure is detected the entire LUOW is rolled back. The definition of failure is application dependent. By default, Shadow Direct terminates (with rollback) any operation that returns a negative SQL Code or an `SQL Code of 100`. Overall this approach allows an application to execute blocks of SQL statements without waiting for the return code from each SQL operation before the next SQL operation is initiated.

Application Development

Some networking environments do not support application development using standard 4GL point-and-click, drag-and-drop, GUI tools such as Visual Basic (VB) and PowerBuilder. For example, MQ (and other messaging products), Wireless, native TCP/IP sockets, and native LU 6.2 all require application developers to create both the client and the server programs and then manage the data stream between them.

This means that the application developers must be completely responsible for the sequence, format, layout, order of the buffers transmitted between the client and the server. The application developers must also handle all error reporting and recovery and in some cases must also handle the data conversions such as ASCII to/from EBCDIC.

Note:
Buffers format and sequencing must be coordinated between the client and server programs. This coordination must be established as the client and server programs are developed, and must be maintained as the client and server programs are modified over time. This can be difficult because it usually is impossible to synchronize the installation of new software releases in a distributed client/server environment.

Shadow Direct solves these problems by providing a message-oriented ODBC driver. This approach allows standard 4GL tools (such as those listed above) to be used with all network environments including messaging systems. The Shadow Direct ODBC driver is then responsible for all of the buffer layout, sequencing, and data conversion issues. Shadow Direct also takes responsibility for maintaining interoperability as the client and server components are independently upgraded.

Note:
Using Shadow Direct also eliminates the need for any host programming in most cases. Instead, the Shadow Server host component takes responsibility for host database connectivity. Overall, Shadow Direct allows application developers to obtain all of the benefits of message systems and other network environments, while eliminating the need for complex network programming needed to use them.

Messages

Traditional clients/system products use permanent TCP/IP sessions or LU 6.2 conversations to connect applications to host systems. This means that these systems can not be used in environments that are hostile towards long-lived network sessions. For example, the Internet does not provide either predictable latency (packet transport times) or even reliability (packet loss rates are sometimes very high). In this kind of environment, long-term network connections will inevitably break leading to client application failures and potential lock conflicts.

Firewalls provide another example of where permanent network connections cannot be used. Many if not most firewalls are configured to prohibit long-term network sessions. This means that traditional client/systems applications typically cannot connect through a firewall.

Shadow Direct solves these problems using message connection mode. In message mode a separate host connection is created and deleted for each message. Message mode provides two major benefits. First, scalability is maximized because network and database resources are only allocated for a very short period of time. Second, a wide variety of standard and non-traditional network transport mechanisms can be used to connect ODBC applications to the host.

Shadow Direct supports message-oriented connections using standard ODBC application programming interfaces. Some operations (such as SELECTs or CALLs) are sent via messages as soon as they are executed. Update operations are accumulated on the client (in Manual-Commit mode) until a LUOW has been completed. The complete LUOW/message is then transmitted to the host and executed. In Auto-Commit mode each update is sent separately to the host using a message.

Shadow Direct supports a variety of network substrates for messages. Each messaging transport has different advantages and disadvantages. The supported messaging systems are:

TCP/IP

Messages can be transmitted using standard TCP/IP and provides all of the scalability and LUOW encapsulation benefits described above. Standard TCP/IP cannot guarantee that a message will be delivered. However, if a message is transmitted successfully messaging does ensure that the LUOW will be handled atomically. Note that TCP/IP also cannot guarantee the delivery of a response message.

LU 6.2

Messages can be transmitted using standard LU 6.2. The pluses and minuses of this approach are essentially the same as TCP/IP (except for the differences between the two protocol stacks).

Wireless

A number of new wireless systems have been deployed recently. Most of these systems are CDPD (Cellular Digital Packet Data) based and support the TCP/IP API. Compared to the Internet they are quite slow (14.4 KB) and may be even less reliable (because of mobile users) from an application standpoint. ODBC messaging addresses many of the problems associated with application deployment using wireless infrastructures.

HTTP

Many companies use firewalls and/or proxy servers to connect their WANs to the Internet or for internal network interconnection. These firewalls and proxy servers are normally configured to block all inbound TCP/IP session requests. They do allow HTTP messages to flow from the Internet (or other parts of the company) to specific servers inside the WAN. Shadow Direct support for HTTP allows ODBC applications to be executed using standard Internet/Intranet facilities without compromising network security by allowing arbitrary inbound TCP/IP session requests.

MQ

The IBM MQ product provides assured message delivery in some cases. Shadow Direct support for MQ allows MQ to be used to reliably deliver ODBC requests to a server. Note that MQ can assure message delivery even across network and host system failures. Shadow Direct support for MQ allows standard 4GL GUI tools (VB and PowerBuilder) to be used to develop MQ applications while retaining all of the advantages of the MQ environment. Note that this approach eliminates the need for the host programming (CICS, IMS, or batch) normally required to use MQ.

Other Messaging products

Several products compete with MQ. Some of the major competitors are NEONet from New Era Of Networks, Pipes from PeerLogic, and Falcon from Microsoft. Shadow Direct supports these products the same way MQ is supported. See “Application Development” on page K-8

APPENDIX L: *Installing a Maintenance Tape*

This appendix provides instructions for installing a Shadow Server maintenance tape.

Unload the CNTL Dataset

Shadow Server’s distribution tape contains 20 libraries for version 2.1 in standard IEBCOPY format. The first library, the CNTL dataset, contains the JCL needed for the rest of the installation process. To unload it, use the following JCL (or some equivalent):

```
//...          JOB
//UNLOAD       EXEC PGM=IEBCOPY
//TAPCNTL      DD   DSN=NEON.CNTL,DISP=(OLD,PASS),
//             UNIT=TAPE,VOL=SER=NSnnnn,
//             LABEL=(1,SL,EXPDT=98000)
//DSKCNTL      DD   DSN=prefix.SV040100.CNTL,DISP=(NEW,CATLG),
//             UNIT=SYSDA,VOL=SER=??????,SPACE=(CYL,(1,1,25)
//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
//
```

Note on the Volume Serial Number

The tape will contain a serial number of the form “Nosing” on its external label. Use this label in the JCL above and in the INSTALL job below.

Backup Considerations

You can either:

- **(Recommended)** Unload the tape into a separate set of libraries.
- Install the tape directly into your production libraries.

Note:

If refreshing an existing set of libraries, be sure to save your already configured Shadow Initialization EXECs (`‘SDBxIN00’`, `‘SWSxIN00’` etc.) in the EXEC or EXECFB library and also any other modified members you wish to save.

Modify the INSTALL Member

Once you have unloaded the CNTL dataset, modify 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 written on the Shadow Server distribution tape.
3. If TAPE is not the correct unit name to use, change the TAPEUNT parameter.
4. Change the DISKPFX parameter to the high-level dataset qualifier that you are using for Shadow Server libraries. The default is SDB.
5. If 3390 cannot be used to refer to the DASD unit on which Shadow Server will reside, change the DISKUNT parameter.

After the Install job has successfully been run, and the Shadow production libraries have been updated, recycle the Shadow started task to put the changes into effect.

We recommend that you rebind the Shadow product plans with the new DBRMs shipped on the maintenance tape. See "Step 4. Bind Shadow Server Product Plans" on page 1-7 of this guide for more information.

After the Install job has been successfully run and the Shadow Server production libraries have been updated, recycle the Shadow Server started task to put the changes into effect.

APPENDIX M:

Setting up Shadow Server to Run under User's TSO Address Space

Setting Up Shadow Server to Run under TSO

Before you can run a Shadow Server under a TSO user's address space, the TSO user must be set up to run exactly as the server. See Chapter 1, "Installing Shadow Server," for more detailed information on implementing the following steps:

1. Allocate all of the Shadow ISPF datasets to the user's Logon Proc:

   Shadow Server:

   ```
   ISPLLIB      NEON.SV040100.LOAD
   ISPMLIB      NEON.SV040100.NEONMLIB
   ISPPLIB      NEON.SV040100.NEONPLIB
   ISPTLIB      NEON.SV040100.NEONTLIB
   SYSEXEC      NEON.SV040100.EXEC(FB) FB if using FB datasets
   SDBTRACE     (optional dd statement see #2 below)
   SDBRPCLB     (optional dd statement see #3 below)
   ```

   Shadow OS/390 Web Server:

   ```
   ISPLLIB      NEON.SV040100.LOAD
   ISPMLIB      NEON.SV040100.NEONMLIB
   ISPPLIB      NEON.SV040100.NEONPLIB
   ISPTLIB      NEON.SV040100.NEONTLIB
   SYSEXEC      NEON.SV040100.EXEC(FB)   FB if using FB datasets
   SWSTRACE     (optional dd statement see #2 below)
   SWSRPCLB(optional dd statement see #3 below)
   ```

2. (optional) Allocate a new trace file. If you do not create a trace file, all trace information is lost during shutdown. To allocate a trace file, use job `'NEON.SV040100.CNTL(DEFDIV)'` to allocate the linear trace dataset. This dataset then needs to be allocated to the user's DD name (`'SDBTRACE'` or `'SWSTRACE'`). It is recommended you only allocate a small trace dataset.

3. In a library allocated to `'SYSEXEC'`, either customize a copy of the existing initialization exec (see "Step 9. Customize Initialization EXEC" on page 1-12) or create a new initialization exec (`'SDBxIN00'` or `'SWSxIN00'`), 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. We recommend that the BROWSEMAX parameter be set at 10000.

4. In the initialization exec, a new port number needs to be defined for TCP/IP connections. For LU6.2 a new APPLID needs to be defined and used.

5. In order to run RPCs allocate your RPC load library to a DDNAME of `‘SDBRPCLB’` or `‘SWSRPCLB’`.

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

Starting a Test Version

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

1. Log on to a TSO/ISPF session.

2. Select the *`Command`* option from the ISPF Primary Options menu.

   The ISPF Command Shell panel appears.

3. Enter the SDB or SWS command followed by the name of the subsystem created in the new initialization exec. This is the one you created when you setup the server to run under TSO. For example:

   ```
   SDB SUB (SDBx)
   ```

   or

   ```
   SWS SUB (SWSx)
   ```

   where `x` is the 4th character of the sysbsystem name.

4. Presss <ENTER>.

   Shadow Server’s or Shadow OS/390 Web Server’s Primary Option Menu appears.

5. Select the *`D`* to run Debug**.**

   Note:
   For more information on using the Debug panel and what the different options do, see the *User’s Guide*.

6. Select *`Test`* from the Debug menu.

7. Enter the `SUBSYSTEM NAME` on the Debugging Control panel.

   The subsystem name is the one you created when you setup the server to run under TSO.

   If you the message:

   - **Shadow Direct**

`The SUBSYS SDBX INITIALIZATION COMPLETE`

- **Shadow OS/390 Web Server**
  `The SUBSYS SWSX INITIALIZATION COMPLETE`

then the server is installed correctly.

8. Return to the Debugging Control panel and stop the server (option **P**).

   If you receive the message:

   - **Shadow Direct**
     `SUBSYS SDBX TERMINATION COMPLETE`
   - **Shadow OS/390 Web Server**
     `SUBSYS SWSX TERMINATION COMPLETE`

   then the server has terminated successfully.

You have verified the installation.

APPENDIX N:
Shadow_VSAM for CICS

This appendix describes how to install the Shadow_VSAM component for CICS. This component allows for the access and update of CICS assigned KSDS VSAM files. For more information about Shadow_VSAM for CICS, see Chapter 11 of the *Shadow Server User's Guide* and Chapter 15 of the *Shadow Web Server User's Guide*.

Prerequisites

Before installing the Shadow_VSAM for CICS, you must have already installed the Shadow Transaction Server for CICS. If you have not done this, refer to Appendix E in the *Shadow Installation Guide* for instructions.

Installation

The installation of Shadow_VSAM for CICS consists of the following steps:

1. Create the CICS resource definitions.
2. Make the `CICSLOAD` dataset modules available to CICS.
3. Restart the CICS region, if necessary.
4. Extract the data map.
5. Verify the Shadow_VSAM for CICS installation.

These steps will be described in detail in the following sections of this appendix.

Note:
The information in Steps 1 through 4 applies to both Shadow Direct and Shadow Web Server. Step 5 has information that is specific to Shadow Direct and to Shadow Web Server, as noted in that step.

Step 1: Create the CICS resource definitions.

Shadow_VSAM for CICS requires several program and transaction resources to be defined in all CICS regions that will be using this component. The Shadow Server CNTL dataset member `CICSCSD` is a sample JCL you can use to help you create these definitions. Refer to this member for further instructions.

Step 2: Make the CICSLOAD dataset modules available to CICS.

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

1. Add the `CICSLOAD` dataset to the `CICS DFHRPL` dataset concatenation.
2. Copy the `CICSLOAD` dataset members to another dataset that is already in the `CICS DFHRPL` dataset concatenation.

The first method is the recommended one for ease of administration, however, if you choose this method, you will need to restart your CICS region.

> ***Note:***
> The `CICSLOAD` dataset should have already been unloaded during the regular Shadow Server installation process, after you submit the INSTALL job to unload Shadow Server's distributed files (see Step 2 in Chapter 1 of the *Shadow Installation Guide*).

Step 3. Restart your CICS region, if necessary.

If, in Step 2, you selected the first method, restart your CICS region before you attempt to use Shadow_VSAM for CICS.

If, however, you selected the second method, you do not need to restart CICS before using the product. In this case, you will need to use the CEDA INSTALL function to make the resource definitions available.

Step 4: Extract Data Map.

You can extract the data map using the Shadow Server Data Mapping Facility. ISPF Option 10.1.8 allows for the extraction of the VSAM file description from a COBOL program listing. For more information about extracting the data map, see Chapter 9 of the *Shadow Server User's Guide*, or Chapter 13 of the *Shadow Web Server User's Guide.*

Step 5: Verify the Shadow_VSAM for CICS installation.

For Shadow Direct users:

1. Make sure you have installed Transaction Server for CICS and have successfully gone through its verification process. If you have already done this, you can skip this step.
2. Check to see that you have the `SDBMAPP` ddname defined to your Shadow server started task JCL. Also, make sure that the map FILEA in the distributed `DATA.MAPS` dataset is in the dataset pointed to by the `SDBMAPP` ddname.

Note:
You can verify this by selecting option 10.2 from the Shadow Server Primary Options Menu, after the Server has been started. Make sure you see FILEA under the structure name column. For more information about invoking the Shadow ISPF interface, see chapter 2 of the Shadow Server User’s Guide.

3. Run the VBDemo as follows*:*

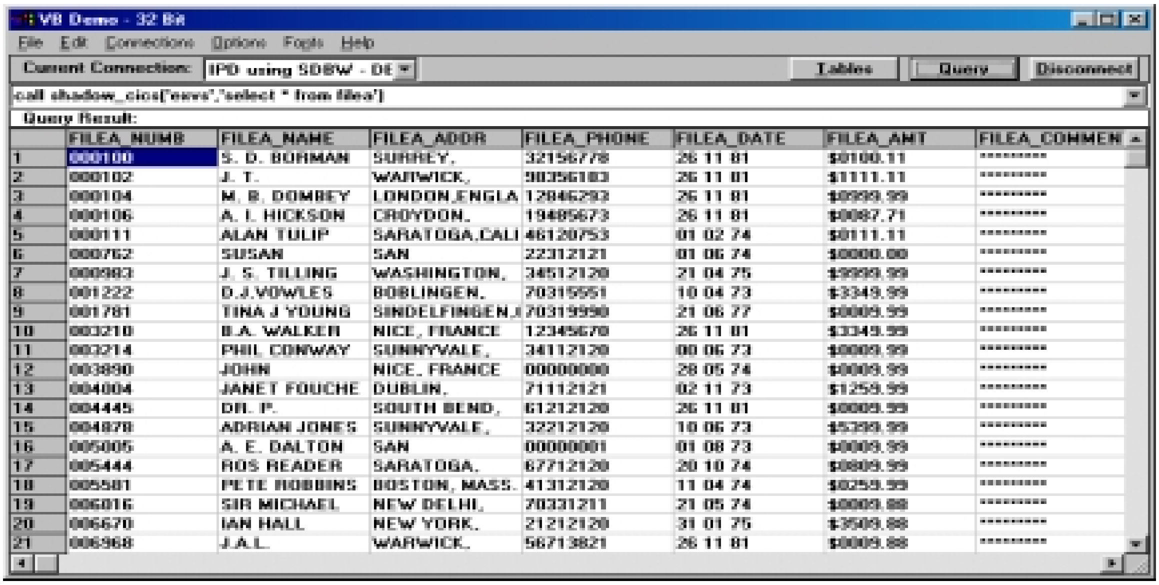
   a. Configure an ODBC data source. For further information, see the Shadow Server User’s Guide chapter, “Configuring Data Sources”.

   b. Bring up the VBDemo distributed program, and connect to the server using the ODBC data source defined above.

   c. Issue the following query:

   ```
   CALL SHADOW_CICS(’EXVS’,’SELECT * FROM FILEA’)
   ```

   d. If the query is successful, you will see data in the VBDemo grid box similar to that shown in Figure N–1:



Current Connection: IPD using SDBW - DE

call shadow_cics('exvs','select * from filea')

Query Result:

	FILEA_NUMB	FILEA_NAME	FILEA_ADDR	FILEA_PHONE	FILEA_DATE	FILEA_AMT	FILEA_COMMEN
1	000100	S. D. BORMAN	SURREY,	32156778	26 11 81	$0100.11	*********
2	000102	J. T.	WARWICK,	98356183	26 11 81	$1111.11	*********
3	000104	M. B. DOMBEY	LONDON,ENGLA	12846293	26 11 81	$0999.99	*********
4	000106	A. I. HICKSON	CROYDON,	19485673	26 11 81	$0087.71	*********
5	000111	ALAN TULIP	SARATOGA,CALI	46120753	01 02 74	$0111.11	*********
6	000762	SUSAN	SAN	22312121	01 06 74	$0000.00	*********
7	000983	J. S. TILLING	WASHINGTON,	34512120	21 04 75	$9999.99	*********
8	001222	D.J.VOWLES	BOBLINGEN,	70315551	10 04 73	$3349.99	*********
9	001781	TINA J YOUNG	SINDELFINGEN,I	70319990	21 06 77	$0009.99	*********
10	003210	B.A. WALKER	NICE, FRANCE	12345670	26 11 81	$3349.99	*********
11	003214	PHIL CONWAY	SUNNYVALE,	34112120	00 06 73	$0009.99	*********
12	003890	JOHN	NICE, FRANCE	00000000	28 05 74	$0009.99	*********
13	004004	JANET FOUCHE	DUBLIN,	71112121	02 11 73	$1259.99	*********
14	004445	DR. P.	SOUTH BEND,	61212120	26 11 81	$0009.99	*********
15	004878	ADRIAN JONES	SUNNYVALE,	32212120	10 06 73	$5399.99	*********
16	005005	A. E. DALTON	SAN	00000001	01 08 73	$0009.99	*********
17	005444	ROS READER	SARATOGA,	67712120	20 10 74	$0809.99	*********
18	005581	PETE ROBBINS	BOSTON, MASS.	41312120	11 04 74	$0259.99	*********
19	006016	SIR MICHAEL	NEW DELHI,	70331211	21 05 74	$0009.88	*********
20	006670	IAN HALL	NEW YORK,	21212120	31 01 75	$3509.88	*********
21	006968	J.A.L.	WARWICK,	56713821	26 11 81	$0009.88	*********

Figure N–1. Shadow Server Query Results

Note:
FILEA is a distributed sample map for the CICS FILEA VSAM file. By default, the map was created using the default connection name EXCS and the default transaction name EXCI. If you have defined a different connection name in your Shadow Server or a different transaction name in CICS, the you need to issue the following call statement:

```
CALL SHADOW_CICS('EXVS','SELECT * FROM FILEA',
'cccc', 'tttt')
```

where `'cccc'` is the connection name defined in your Shadow Server, and `'tttt'` is the transaction name defined in CICS. The transaction name **must** always point to the DFHMIRS program.

For Shadow Web Server users:

1. Make sure you have installed Transaction Server for CICS and have successfully gone through its verification process. If you have already done this, you can skip this step.

2. Check to see that you have the `SWSMAPP` ddname defined to your Shadow Web Server started task JCL. Also, make sure that the map FILEA in the distributed `DATA.MAPS` dataset is in the dataset pointed to by the `SWSMAPP` ddname.

   Note:
   You can verify this by selecting option 10.2 from the Shadow Server Primary Options Menu, after the Server has been started. Make sure you see FILEA under the structure name column.

3. Using a browser from the client, enter your URL to display the Shadow Web Server home page.

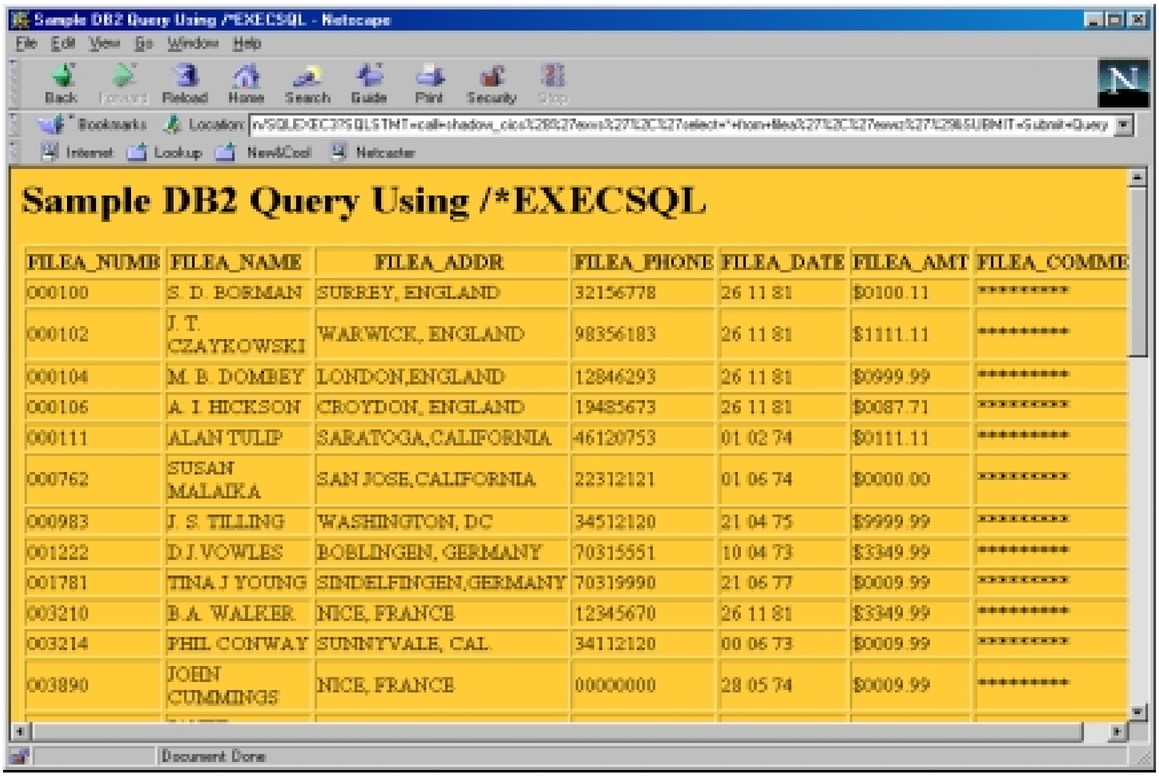
4. On the home page, click Sample Transactions.

5. In the sample applications, select option 5, “Illustrate the use of EXECSQL.”

6. Submit the following query:

   ```
   CALL SHADOW_CICS('EXVS','SELECT * FROM FILEA')
   ```

   If the query is successful, you will see data similar to that shown in Figure N–2:


Sample DB2 Query Using /*EXECSQL - Netscape
File Edit View Go Window Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks Location:
Internet Lookup New&Cool Netcaster
Document: Done

n/SQLEXEC3?SQLSTMT=call+shadow_cics%28%27exvs%27%2C%27select+*+from+filea%27%2C%27exvs%27%29&SUBMIT=Submit+Query

Sample DB2 Query Using /*EXECSQL

FILEA_NUMB	FILEA_NAME	FILEA_ADDR	FILEA_PHONE	FILEA_DATE	FILEA_AMT	FILEA_COMME
000100	S. D. BORMAN	SURREY, ENGLAND	32156778	26 11 81	$0100.11	**********
000102	J. T. CZAYKOWSKI	WARWICK, ENGLAND	98356183	26 11 81	$1111.11	**********
000104	M. B. DOMBEY	LONDON,ENGLAND	12846293	26 11 81	$0999.99	**********
000106	A. I. HICKSON	CROYDON, ENGLAND	19485673	26 11 81	$0087.71	**********
000111	ALAN TULIP	SARATOGA,CALIFORNIA	46120753	01 02 74	$0111.11	**********
000762	SUSAN MALAIKA	SAN JOSE,CALIFORNIA	22312121	01 06 74	$0000.00	**********
000983	J. S. TILLING	WASHINGTON, DC	34512120	21 04 75	$9999.99	**********
001222	D.J.VOWLES	BOBLINGEN, GERMANY	70315551	10 04 73	$3349.99	**********
001781	TINA J YOUNG	SINDELFINGEN,GERMANY	70319990	21 06 77	$0009.99	**********
003210	B.A. WALKER	NICE, FRANCE	12345670	26 11 81	$3349.99	**********
003214	PHIL CONWAY	SUNNYVALE, CAL.	34112120	00 06 73	$0009.99	**********
003890	JOHN CUMMINGS	NICE, FRANCE	00000000	28 05 74	$0009.99	**********

Figure N–2. Shadow OS/390 Web Server Query Results

Note:

FILEA is a distributed sample map for the CICS FILEA VSAM file. By default, the map was created using the default connection name EXCS and the default transaction name EXCI. If you have defined a different connection name in your Shadow Web Server or a different transaction name in CICS, you need to issue the following call statement:

```
CALL SHADOW_CICS('EXVS','SELECT * FROM FILEA',
'cccc', 'tttt')
```

where `‘cccc’` is the connection name defined in your Shadow Web Server, and `‘tttt’` is the transaction name defined in CICS. The transaction name **must** always point to the DFHMIRS program.

APPENDIX O: Recoverable Resource Manager Services Attachment Facility (RRSAF) Support

RRSAF is a new connection method provided by DB2 v5.1. Shadow Direct v 4.5 and Shadow Web Server v 4.5 support it, as well as the older CAF (Call Attach Facility), the method used by Shadow products in earlier releases. This documentation describes:

- What is RRSAF and how it works.
- Guidelines to determine when to use RRSAF with Neon Products.
- What you will need in order to configure RRSAF at your site.
- Extending OS/390 RRS to BEA Tuxedo and Microsoft Transaction Server environments.

For information on configuring and activating RRSAF support, refer to "Step 28. (optional) Activating RRSAF and Two Phase Commit Support" on page 1-26 of Chapter 1 in the Shadow Installation Guide.

What It Is

RRSAF allows an MVS address space to connect to DB2. Significantly, it works with the RRMS component of MVS, allowing updates to a DB2 system to become part of a transaction managed by RRMS. With the Shadow support of RRMS and distributed transactions available with Shadow Direct 4.5, transactions originating on the Neon client side can now involve updates to DB2 on MVS (as well as other MVS data sources and transactional resource), in addition to other data sources off the mainframe. Shadow on the MVS side functions as a local transaction coordinator, communicating with the client side transaction manager (typically Microsoft MTS or BEA Tuxedo) to allow distributed transactions using two-phase commit protocol.

CAF (Call Attach Facility)

Before RRSAF, CAF was the primary method provided by DB2 for a non-IBM address space to connect to DB2. CAF is still supported by DB2 and Shadow; however it does not provide support for distributed transactions.

Choosing an Attachment Facility

Shadow makes the determination of which attach facility to use for connecting to DB2 (CAF or RRSAF) at initialization time, based on the setting of the "DB2ATTACHMENTFACILITY" product parameter. The options are "CAF" or

“RRS” with the default being CAF. The attach facility method can't be changed after initialization.

Guidelines to determine when to use RRSAF

1. If you are using CAF and have a virtual storage shortage below 16MB, try RRSAF. It uses more than 2k less per user than CAF and could free up the necessary storage (potentially over 2 meg below 16MB storage when over 1000 DB2 threads are in use).
2. If you wish to do distributed two-phase commit transactions using Shadow, with DB2 as one of the data sources, you must choose RRSAF as the Shadow attachment method.
3. If only a small percentage of DB2 updates need two-phase-commit transaction support, consider creating a separate copy of Shadow for those, and use RRSAF with that copy.

Configuring RRSAF at your Site

In order to run RRSAF at your site, you will need:

- **DB2 at version 5.1 and later.** If a single copy of Shadow must also access a DB2 system at an earlier release, you cannot use RRSAF with that copy of Shadow.
- **Shadow Server v 4.5 or higher.** The server can use either CAF or RRSAF, but not both. You must choose the method at startup. It cannot be changed without restarting the server.
- **OS/390 RRS installed and running.** If RRS is not running, all RRSAF requests are rejected by DB2.
- **RRS requires System Logger Log Streams**, which can use the Coupling Facility or be DASD-only. To use DASD-only Log Streams, you must be at OS/390 release 2.4 or later, and put on the following two PTFs: UW43929 and UW43930. (Before OS/390 release 4, the System Logger did not support DASD-only Log Streams).

Setting up and Running RRS

The best source for understanding what is needed to set up and run RRS is the IBM manual "*MVS Programming: Resource Recovery*" (GC28-1739), chapter 8 "Managing RRS". This chapter refers to the IBM manual "Setting Up a Sysplex", for instructions on how to setup and prepare System Logger Log Streams.

Extending OS/390 RRS to BEA Tuxedo and Microsoft Transaction Server Environments

The Shadow Resource Recovery System ("RRS") component allows the extension and coordination of the protected resources between the OS/390 Shadow environment, for IMS and DB2 resources, and the BEA WebLogic ("WebLogic") and Microsoft Transaction Server ("MTS") environments. This allows an application program to make consistent changes to multiple resources.

A common example for this usage is a banking application, in which a customer wishes to transfer a certain amount of money from his savings account to his checking account. Either both updates must be made or neither of them made. The technical terminology for this functionality is Two Phase Commit or 2PC, and the resources protected in this scheme are called protected or recoverable resources. Protected resources must hold their integrity in case of hardware or software failures, communication failures, human errors or a catastrophe.

BEA supplies a similar function in their Enterprise WebLogic (formerly Tuxedo) product and, similarly, Microsoft supplies the same functionality in their Microsoft Transaction Server product. The standard for this functionality is defined in the XA Specification from the X/Open organization's Common Application Environment.

For this particular function, Shadow is the Server and either WebLogic or MTS is the client. Shadow takes on the role of the Server Distributed Syncpoint Manager ("SDSRM") taking direction from either WebLogic or MTS and relaying and coordinating this information with the OS/390 Resource Recovery System. Updating OS/390 IMS and DB2 protected resources will then be coordinated with updates of protected resources on the client systems.

To activate and control RRS, the following parameters may be specified:

- **DB2ATTACHFACILITY** must be set to RRSAF.
- **RRS** must be set to YES to activate RRS.
- **RESOURCEMGRNAME** must be set to the unique sysplex name of the RRS Resource Manager, which is a Server Distributed Syncpoint Manager (SDSRM). If a default value is not specified, product initialization creates a 32 character name as follows:

  Chars 1-24: NEONRRS.RESOURCE.MANAGER
  Chars 25-28: The Shadow Subsystem name such as SDBA, SDBB, etc.
  Chars 29-32: System SMF ID

  If the name is changed after the system is operational, any incomplete (in-doubt) transactions from the previous run will not be able to be completed.
- **RRSCONTEXT** should be set to either NATIVE or PRIVATE, depending on the RRS context.

- **RECTABLEENTRIES** must be set to specify the maximum number of entries in the RRS recovery table. The range is from 200 to 400, with 400 being the default value. If the maximum size of the table is exceeded, information on in-doubt transactions will be lost.
- **TRACERRSEVENTS** must be set to YES if you want RRS events to be traced via the Trace Browse Facility.
- **TRACEFULLRRSDATA** must be set to NO if you do not want the entire RRS workarea to be traced for RRS events using Trace Browse. When set to NO, only the amount of data that will fit in a standard message block will be traced.
- **CLEARARCHIVERECOVERY** must be set to NO if you do not want any in-flight archive recovery and cleanup operations to be bypassed during startup. Cleanup of an incomplete trace browse archive must be handled manually, since setting this flag to YES causes the RRS Server to delete all information needed to invoke automatic cleanup at a later time.

Note:

For more information about each one of these parameters, please see the Started Task Parameters section in the Shadow Server User’s Guide. This information includes a complete description of each parameter, its default value, whether or not it is modifiable after initialization, and whether or not it is output only.

APPENDIX P: *AutoHTML - Web Enabling Transactions (SWS)*

This appendix describes the procedure for executing online IMS transactions and commands.

Implementation Overview

Shadow Web Server uses IMS/APPC in order to execute online IMS transactions and commands. In order to use this interface, you must:

1. Configure your IMS system to support IMS/APPC.
   - Configure IMS/APPC.
   - Install the IMS LU 6.2 User Edit Exit (`DFSLUEE0`).
2. Configure MVS/APPC Support within Shadow Web Server MVS/APPC Prerequisites.
3. Configure Shadow Web Server for IMS transactions.
4. Enable IMS Transaction Server.
5. Verify the Installation.

Web Enabling Transactions

Once you have the APPC interface configured, Web enabling your IMS transactions are as simple as:

1. Generate input and output transaction data format maps.
2. Generate HTML to display the output data on the Web Browser.
3. Build a rule to process your transaction.

Note:

Shadow Web Server’s Mapping Facility provides a conversion utility that will convert your MFS Source to the required format maps as well as generate an HTML page in the image on your system.

Now you can assign LTERMs to IMS Transactions.

IMS/APPC Interface

Step 1. Configure IMS System to Support IMS/ APPC

Configuring IMS/APPC

IMS Prerequisites

Before you can use the Transaction Server for IMS, you must first configure IMS/ APPC by performing the following steps. Please refer to the *IMS Data Communication Administration Guide* for additional details.

- You must be using IMS 4.1 or above.
- The system parameter in the `'IMSCTRL'` macro must specify a MVS version greater than or equal to 4.2, for 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
```

- To start APPC without restarting IMS, issue the following command: **/START APPC**
- `SET APPC=YES` in the startup definitions for your IMS DCCTL region. Depending on your installation, this could be member `'DFSPBIV1'` in your IMS Proclib dataset.

Step 2. 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 Shadow Web 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`).

Shadow Web Server delivers a load module for the IMS LU 6.2 User Edit Exit. This load module may be found in Shadow Web Server’s load library `‘hlq.LOAD(DFSLUEE0)’`. JCL to relink this exit into your IMS `‘RESLIB’` may be found in Shadow Web Server’s sample library `‘hlq.SAMP(JCLLUEE0)’`.

Once you’ve relinked the exit into your IMS `‘RESLIB’`, you will need to stop and restart your IMS system for the exit to take effect.

Configure MVS/APPC Support within Shadow Web Server

MVS/APPC Prerequisites

Define the IMS APPC LU and the additional local LU to MVS/APPC in `‘SYS1.PARMLIB(APPCPMxx)’`:

```
/**********************************************************/
/* 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)
```

VTAM Prerequisites

Define the IMS APPC APPLID and an additional Local APPC LU to VTAM. For example:

```
SDBIMS   VBUILD TYPE=APPL
IMSLU62 APPL   ACBNAME=IMSLU62,  ACBNAME FOR APPC/IMS        +
            APPC=YES,                                       +
            AUTOSES=0,                                      +
            DDRAINL=NALLOW,                                 +
            DLOGMOD=APPCHOST,                               +
            DMINWNL=5,                                      +
            DMINWNR=5,                                      +
            DRESPL=NALLOW,                                  +
            DSESLIM=10,                                     +
            LMDENT=19,                                      +
            PARSESS=YES,                                    +
            SECACPT=CONV,                                   +
            SRBEXIT=YES,                                    +
            VPACING=0
MVSLU02 APPL   ACBNAME=MVSLU02,   ACBNAME FOR APPC           +
            APPC=YES,                                       +
            AUTOSES=0,                                      +
            DDRAINL=NALLOW,                                 +
            DLOGMOD=APPCHOST,                               +
            DMINWNL=5,                                      +
            DMINWNR=5,                                      +
            DRESPL=NALLOW,                                  +
            DSESLIM=10,                                     +
            LMDENT=19,                                      +
            MODETAB=APPCTAB,                                +
            PARSESS=YES,                                    +
            SECACPT=CONV,                                   +
            SRBEXIT=YES,                                    +
            VPACING=0                                       +
```

Configure Shadow Web Server for IMS Transactions

- **Specify the “`APPC/IMS`” parameter** in Shadow Web Server’s initialization member (`‘SWSxIN00’`)). For example:

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

- **Activate the IMS Exit.** Once you have the IMS LU 6.2 User Edit Exit (`DFSLUEE0`) installed in the running IMS, you must activate the usage of this exit within the Shadow Server address space. This is done with the following command:

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

> ***Important:***
> Shadow Server adds data to the messages sent to the IMS System if this value is specified. The NEON-supplied exit removes the additional data so the IMS transaction functions normally.

- **Set Default Parameters.** It is also recommended that you establish a default IMS/APPC Partner LUNAME and the default Security Level.

The IMSSECURITYTYPE has three options:

- **NONE.** No security information is passed to IMS/APPC during the execution of the transaction or command. This may be acceptable for non-production IMS environments or environments where security is controlled through application program interfaces rather than IMS system interfaces (AGN, RACF, CA-ACF2, etc.).
- **SAME**. Uses the security information associated with the user who signed onto the server and passes it along with the IMS transaction. IMS authorizes the transaction or command based upon IMS system level security established during the IMS generation process.
- **PROGRAM.** Indicates the userid and password are received from the executing program. With enabled IMS transactions, this is controlled by a parameter in the `/*EXECIMS` section of the `/*WWW` rule. For example:

```
“MODIFY PARM NAME(IMSSECURITYTYPE)      VALUE(SAME)”
```

The IMSPARTNERLU parameter is the IMS APPC LUNAME which is created when you configured your IMS system to support IMS/APPC. For example:

```
“MODIFY PARM NAME(IMSPARTNERLU)   VALUE(P390.P392AIMS)”
```

Enable IMS Transactions

Step 1. Generate Format Maps from MFS Source

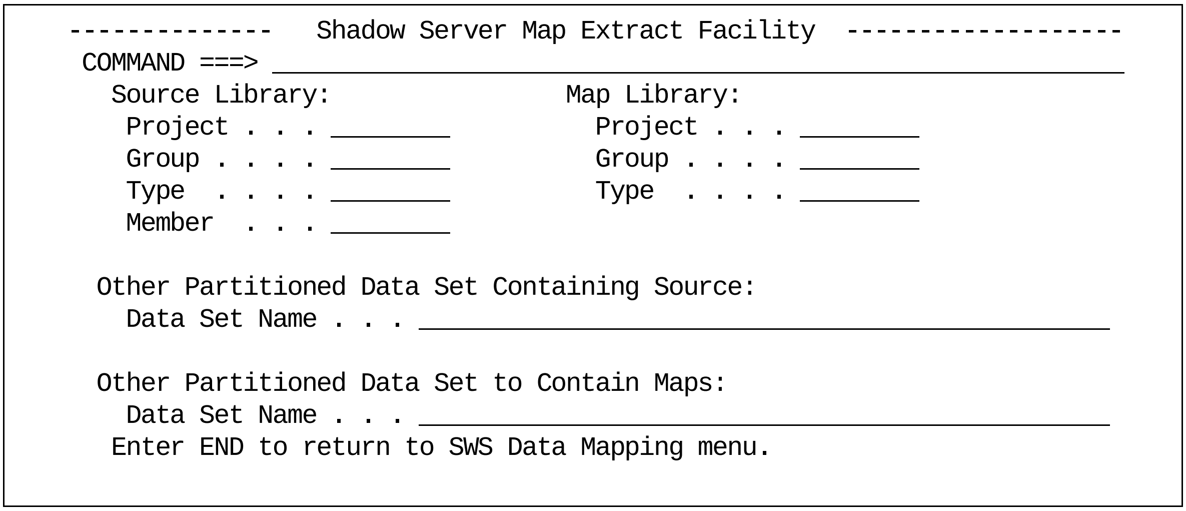
Before using Shadow Web Server’s Mapping Facility, you must know the following:

- **The name of your map dataset.** This is the dataset assigned to the **“**`SWSMAPP`**”** ddname in Shadow Web Server’s startup JCL.
- **The name of your MFS Source library.** Contact your IMS System Administrator if you do not know the name of this library.

In order to generate the input and output format maps from your MFS source, select:

- The Data Mapping Facility from Shadow Web Server’s Primary Option Menu.
- The EXTRACT option from Shadow Server’s Mapping Facility.
- The Extract MFS option from the Shadow Server’s Mapping Facility.

Below is an example of the ISPF interface to the MFS Extract Facility. You need to enter your MFS Source library name as the “`Source Library`” and Shadow Web Server’s Map dataset name as the “`Map Library`”.



```
 -------------   Shadow Server Map Extract Facility  ------------------
 COMMAND ===> __________________________________________________________
   Source Library:               Map Library:
    Project . . . ________         Project . . . ________
    Group . . . . ________         Group . . . . ________
    Type  . . . . ________         Type  . . . . ________
    Member  . . . ________

  Other Partitioned Data Set Containing Source:
    Data Set Name . . . _______________________________________________

  Other Partitioned Data Set to Contain Maps:
    Data Set Name . . . _______________________________________________
   Enter END to return to SWS Data Mapping menu.
```

If you don't specify a member name, a member list will be displayed for you to select a member. Each member you select will be parsed into a data map and stored in the data mapping facility.

Once you have extracted all the members you want to, press <PF3> (two times) in order to back up to the Shadow Server Mapping Facility Menu. At this screen, you need to select the `Map Refresh` option. This makes your format maps available to Shadow Web Server.

Essentially, the extract generates the equivalent of a `Message Input Descriptor` (MID), `Message Output Descriptor` (`MOD`) and a `Device Output Format` (`DOF`).

The `Input Map` (or `MID`) is used to format data from the HTML page for input to a specific transaction or command. Each HTML page must contain a query variable named “`INPUTMAP`” which points to an extracted map. The `Input Map` has a pointer to the `Output Map`. By default, this `Output Map` will be used to parse the transaction output. If you install the NEON supplied IMS LU 6.2 User Edit Exit (`DFSLUEE0`), the `Output Map` (`MOD`) specified by the IMS transaction will be used instead.

The `Output Map` (or `MOD`) is used to parse the output from the IMS transaction and build SQL Column Names. These column names are the same as those specified in the MFS Source. The `Output Map` contains a pointer to the Output Screen.

The `Output Screen` (or `DOF`) is used to define the literals displayed in the HTML page as well as the placement of the SQL Column data supplied by the online IMS transaction. The Output Screen Map contains the name of the associated HTML dataset and member.

Step 2. Generate HTML from MFS Source

The objective of this facility is to generate HTML in the image of your output MFS screen. This is accomplished by formatting the HTML from the "`DEV`" (or `Device Output Format (DOF)`) portion of the MFS Source. This is the Output Screen Map.

Before using Shadow Web Server's Mapping Facility, you must know the following:

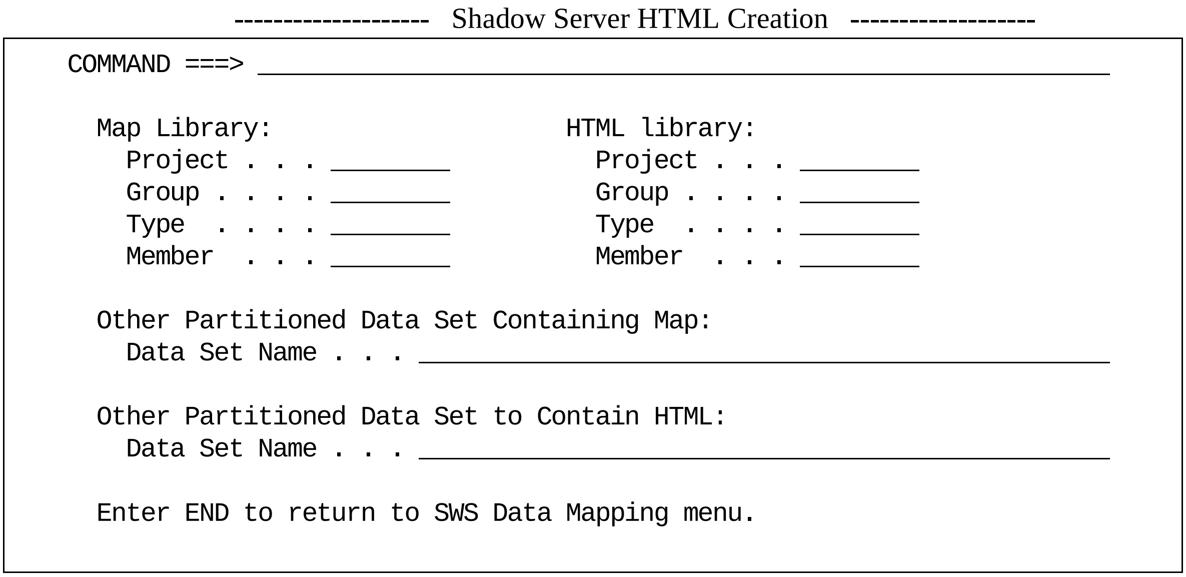
- **The name of your map dataset.** This is the dataset assigned to the `"SWSMAPP"` ddname in Shadow Web Server's startup JCL.
- **The name of your output HTML Source library.** You may create your own dataset with the following specifications:
  - Record Format is VB
  - Record Length is 19036
  - Block Length is 19040
  - Dataset Organization is PO

In order to generate HTML from your format maps, select:

- The Data Mapping Facility from Shadow Web Server's Primary Option Menu.
- The `Gen HTML` option from Shadow Server Mapping Facility.

The following is an example of the ISPF interface to the Shadow Server HTML Creation utility. You need to enter:

- Your HTML Source library name as the "HTML Library".
- Either Shadow Web Server's (perm) map dataset name or your working (`temp`) one as the "`Map Library`".



```
-------------------  Shadow Server HTML Creation  -------------------
 COMMAND ===> _________________________________________________________

   Map Library:                   HTML library:
     Project . . . ________         Project . . . ________
     Group . . . . ________         Group . . . . ________
     Type  . . . . ________         Type  . . . . ________
     Member  . . . ________         Member  . . . ________

   Other Partitioned Data Set Containing Map:
     Data Set Name . . . _______________________________________________

   Other Partitioned Data Set to Contain HTML:
     Data Set Name . . . _______________________________________________

   Enter END to return to SWS Data Mapping menu.
```

If you don't specify a member name, a member list will be displayed for you to select a member. Each member you select will be parsed into a data map and stored in the data mapping facility.

Once you have extracted all the members you want press <PF3> (two times) in order to back up to the Shadow Server Mapping Facility Menu. At this screen, you need to select the `Map Refresh` option. This makes your format maps available to Shadow Web Server.

Step 3. Build a /*EXECIMS rule

If you need special processing for your IMS transaction, you may build a `/*WWW` rule with a `/*EXECIMS` section.

The `/*EXECIMS` section expects the input URL to contain query variables that match the names of field names within the Input Map. It also expects to have a query variable (INPUTMAP) passed on the input URL indicating which Input Map to use in order to build the input transaction.

NEON Systems ships a standard IMS entry point named:

```
.../NEON/IMS
```

This entry provides an interface that allows the user to enter an IMS transaction or IMS command.

Pressing ENTER on this page will cause another Neon supplied rule to be started which will run your IMS commands and transactions. The second rule, also supplied by neon, is in:

```
.../NEON/IMSENTRY
```

With these two rules, the IMS LU 6.2 User Edit Exit (`DFSLUEE0`) installed and HTML generated by Shadow Web Server's Mapping Facility, you should be able to run the majority of your IMS transactions.

Restrictions

No output logical paging. Shadow Web Server IMS Transaction Server only supports physical page output.

Step 4. Generate the IMS Default HTML

IMS Ships several default MFS screens. Most of these are used to display system data upon various types of devices. Some of these are used as defaults for outbound messages where the transaction did not specify a MODNAME. NEON has supplied default source that **must** be run through the MFS Extract and HTML Generation process before any IMS commands or transactions may be executed. The following source needs to be extracted:

```
hlq.SAxMP(IMSENTRY)
hlq.SAMP(IMSMOD)
```

Once extracted, the following `DOF` names must have HTML generated:

```
DFSDF2
DFSDOF1
```

Verify the Installation

Restart IMS, APPC, and ASCH. You should see the following after issuing the IMS startup commands:

```
DFS1960I IMS HAS REQUESTED A CONNECTION WITH APPC/MVS   IVP1
DFS1960I IMS HAS REQUESTED A CONNECTION WITH APPC/MVS   IVP1
```

followed by:

```
 ATB050I LOGICAL UNIT IMSLU62 FOR TRANSACTION SCHEDULER  IVP1 HAS BEEN
ADDED TO THE APPC CONFIGURATION.
DFS1958I IMS CONNECTION TO APPC/MVS COMPLETE, LUNAME=xxxxxxxx  IVP1
```

You can now use the following NEON-supplied URL to access your IMS system.

…/NEON/IMS

When you enter this URL on a web browser, you will get a framed display where the left frame represents the keyboard and the right frame represents the screen image. Enter the IMS Commands or Transaction Codes in the right frame and press <ENTER>.

Try a simple display command, **/DIS A** for instance.

Once you type in the command or transaction code and press <ENTER>, you get a "pop-up" dialog box which asks for your Userid and Password. These are used to validate your access to the MVS system that Shadow Server is running. After your Userid and Password have been validated, the command or transaction code you entered is passed to IMS and the is response returned to you.

IMPORTANT:

With the NEON-supplied IMS LU 6.2 User Edit Exit (`DFSLUEE0`) in-place, Shadow Server follows the MID-MOD chaining. This means that transactions that message switch are recognized by Shadow Server and the HTML associated with the output `MOD` is sent. If the output MFS has not been extracted, you will receive an `Error Code 47`. If you have not generated the HTML, you will receive an error message of `UNABLE TO LOCATE OUTPUT FORM`.

Assign LTERMs to IMS Transactions

IMS provides the application program with an I/O PCB containing information about with whom the application program is interacting. This control block contains the LTERM (Logical Terminal) name of the terminal, Userid of the person who signed on and the MODNAME of the input message. NEON automatically updates the Userid and MODNAME information. Unfortunately, a distributed IP network does not provide a mechanism for capturing an LTERM-equivalent name, but with this feature, you can now assign meaningful LTERM names based upon Userid’s or TCP/IP Addresses.

Purpose

When an IMS transaction is passed to Shadow Web Server, the IMS Transaction Server scans a table looking for a matching Userid or TCP/IP address provided the IMSLTERMTABLESEQ parameter (PRODIMS group) is set to USERID or IPADDRESS.

Note:

Even with IMS LTERM Table entries present, the parameter must be set to USERID or IPADDRESS, otherwise a scan is not performed. The default for the IMSLTERMTABLESEQ parameter is NONE.

If a matching entry is found, the associated LTERM is passed to the NEON-Supplied IMS Exit (`DFSLUEE0`), which places the entry into the I/O PCB for use by the IMS application program.

The IMS LTERM Table is maintained in Extended Private storage within the server’s address space. Entries in the table can contain generic Userids and TCP/IP addresses. Once added, these entries can be enabled or disabled at any time. The changes are effective immediately.

Many IMS application programs use the LTERM to determine things such as output printer destination and application-based security.

How Does it Work?

To use the IMS Control Facility, select the “IMS” option from the Shadow Server’s Primary Option ISPF Panel. The IMS Control Facility allows you to:

- Display the current selection table.
- Modify the current selection table.
- Add new entries to the table.
- Copy the table to a sequential file.

IMS LTERM Table entries are added, modified and displayed using **DEFINE, MODIFY** and **DISPLAY** commands. These commands may be entered via a REXX exec (under ADDRESS SDB or SWS) such as `‘SDBBIN00’` or `‘SWSSIN00’`.

Changing entries

The IMS Control Facility Change screen provides a list of all the table entries in the server, which is initially displayed in LTERM sequence. This list can be sorted into Userid, TCP/IP Address and LTERM sequence with a SORT command followed by a sequence parameter of USERID, IPADDR or LTERM.

Once the list is displayed, any entry in the list can be modified by doing the following:

1. Overtype the LTERM field.
2. Press <ENTER>.

Multiple entries may be modified before pressing <ENTER>.

Adding Table Entries

The IMS Control Facility Add screen provides the ability to add new entries to the IMS LTERM table in the server. These are dynamic in nature and disappear when the server is “bounced”. The only permanent entries are those added through DEFINE commands in the `‘SWSSIN00’` or `‘SDBBIN00’` specification.

Creating a File of the current IMS LTERM Table

The IMS Control Facility File screen, provides the ability to create a file of “**DEFINE**” commands for all the IMS LTERM Table entries currently stored in the server. This allows user's to add table entries and create a file that can be added to, or run from, `‘SWSSIN00’` or `‘SDBBIN00’`.

Shadow Web Server and Shadow Direct

This works with both Shadow Web Server and Shadow Server.

- **Shadow Web Server.** It is available for the express purpose of supporting IMS/TM Web Enablement.

- **Shadow Direct.** The features and functions of the tables and the associated IMS Exit (DFSLUEE0) can be exploited through the CALL Shadow_IMS interface.

Note:
Help screens are available by pressing <PF1>.

Format

The format of the commands are:

```
DEFINE IMSLTERM USERID(xxxxxxxx) IPADDRESS(xxxxxxxxxxxxxxx)
LTERM(xxxxxxxx) STATUS(ENABLE)

MODIFY IMSLTERM USERID(xxxxxxxx) IPADDRESS(xxxxxxxxxxxxxxx)
LTERM(xxxxxxxx) STATUS(DISABLE)

DISPLAY IMSLTERM USERID(xxxxxxxx) or DISPLAY IMSLTERM
IPADDRESS(xxxxxxxxxxxxxxx) STATUS(ENABLE)
```

Index

Reader’s Comment Form

At NEON Systems, Inc. we are always looking for good ideas. If you have a suggestion or comment regarding any of our publications, please complete this form, and mail or fax it to us at the following address. Thank you.

Please complete the following information, or attach your business card here.

Your Name: ______________________________

Phone Number: ______________________________

Your Company: ______________________________

Address: ______________________________

____

Publication Name: ______________________________

Version *and* Edition Numbers (see page ii)**:** ______________________________

Suggestion/Request: ______________________________

____

____

____

____

____

____

Please mail or fax this page to:

NEON Systems, Inc.
14100 SW Freeway, Suite 500
Sugar Land, Texas 77478, U. S. A.

Fax Number: **(281) 242-3880**