

OS/390



SOMobjects Configuration and Administration Guide

OS/390



SOMobjects Configuration and Administration Guide

Note

Before using this information and the product it supports, be sure to read the general information under "Notices" on page vi.

Second Edition, September 1997

This is a major revision of GC28-1851-00.

This edition applies to OS/390 Version 2 Release 4 (5647-A01) and to all subsequent releases and modifications until otherwise indicated in new editions.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

IBM welcomes your comments. A form for readers' comments may be provided at the back of this publication, or you may address your comments to the following address:

International Business Machines Corporation
Department 55JA, Mail Station P384
522 South Road
Poughkeepsie, NY 12601-5400
United States of America

FAX (United States & Canada): 1+914+432-9405
FAX (Other Countries): Your International Access Code +1+914+432-9405

IBMLink (United States customers only): KGNVMC(MHVRCFS)
IBM Mail Exchange: USIB6TC9 at IBMMAIL
Internet e-mail: mhvrdfs@vnet.ibm.com
World Wide Web: <http://www.s390.ibm.com/os390>

If you would like a reply, be sure to include your name, address, telephone number, or FAX number. Make sure to include the following in your comment or note:

- Title and order number of this book
- Page number or topic related to your comment

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1995, 1997. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	vi
Examples in This Book	vi
Programming Interface Information	vi
Trademarks	vi
About This Book	ix
Who Should Use This Book	ix
How This Book Is Organized	ix
Where to Find More Information	x
Related SOMobjects Publications	x
Summary of Changes	xi
Chapter 1. Configuring Your SOMobjects Environment	1
Before You Begin	1
Getting Started with Non-Distributed SOMobjects	1
Specifying OS/390 Initialization Settings for the SOM Subsystem	2
Creating a JCL Procedure for the SOM Subsystem	3
Defining a Security Identity for the SOM Subsystem	4
Specifying Configuration File Values for the SOM Subsystem	4
Starting the SOM subsystem	5
Validating Your Configuration by Running a Non-Distributed Application	6
Getting Started with Distributed SOMobjects (DSOM)	7
Creating a JCL Procedure for SOMobjects Servers	7
Defining a Security Identity and Authorizations for SOMobjects Servers	7
Registering SOMobjects Servers with WLM	7
Enabling WLM Server Management	8
Running the SOMobjects Configuration Utility (SOM@CFG)	9
Recycling the SOM subsystem	9
Configuring and Running an Application Server	10
Creating a JCL Procedure for the Application Server	10
Defining Security Identity and Authorization for the Application Server	11
Defining the Application Server to WLM	11
Registering the Application Server with the SOM subsystem	11
Capacity Planning for Running Many Distributed Clients	12
Validating Your Configuration by Running a Distributed Application	13
Chapter 2. Configuring and Administering SOMobjects with Workload Manager (WLM)	15
Defining Servers to Workload Manager (WLM)	15
Defining the Application Environment for the Naming Server	17
Defining the Application Environment for the Security Server	22
Defining an Application Server (CARSERVER1) to WLM	26
Managing Performance with Workload Manager (WLM)	30
Creating a Workload	31
Creating a Service Class	32
Creating Classification Rules	36
Classifying Work by User ID	38
Classifying Work by Class Name	39
Classifying Work by Method Name	40

Classifying Work by Application Environment Name	41
Creating a Service Policy	44
Installing a Definition	47
Activating the Service Policy	48
Managing Servers Using WLM Commands	51
Displaying a Server's Status	51
Stopping a Server	51
Making a Server Available	51
Stopping and Making a Server Available	51
Chapter 3. Registering Classes and Servers in the Implementation Repository	53
Using the REGIMPL Utility to Register Servers	53
Running REGIMPL in Prompting Mode	54
Adding a Server to the Implementation Repository	54
Changing a Server in the Implementation Repository	56
Show One Server in the Implementation Repository	57
Show All Servers in the Implementation Repository	58
List Aliases	58
Adding a Class to a Server Implementation	59
Adding a Class to all Servers	60
Listing Classes in the Implementation Repository	60
Deleting a Class from All Server Implementations	60
Deleting One Class from a Server Implementation	61
Deleting a Server in the Implementation Repository	61
Saving Your Data and Exiting REGIMPL	62
Running REGIMPL in Fast Path Mode	62
REGIMPL Help Screen	62
REGIMPL Command Syntax	63
REGIMPL Example 1: To Add a Server	65
REGIMPL Example 2: To Add a Class to a Server	65
Chapter 4. Configuring and Administering Security for SOMobjects	67
Before You Begin	68
Auditing Considerations	68
Security Configuration for All SOMobjects Installations	68
Associating User IDs with the SOM Subsystem and SOMobjects Servers	69
Authorizing SOMobjects Servers to the SOM Subsystem	70
Authorizing SOMobjects Servers and Clients to OpenEdition	70
Running a Server in Secure Mode	71
Configuring a Server to Run in Secure Mode	71
Authorizing Clients to Secure Servers	71
Authorizing Clients to SOMobjects	72
Setting up the PassTicket Environment	72
Authorizing Clients to Specific Methods	73
Creating Profiles in the GSOMDOBJ Class	74
Chapter 5. Setting up Configuration Files	75
About Configuration Files	75
Configuration File Variable Settings	75
Comparison of Global and Local Configuration File Values	76
Configuration File Syntax	77
Stanzas and Keyword Variables	77
[sombt] Stanza (BTree Framework)	77

[some] Stanza (SOMobjects Compiler)	78
[somed] Stanza (Distributed SOMobjects)	80
[SOMD_TCPIP] Stanza (TCP/IP Protocol)	82
[somir] Stanza (Interface Repository)	83
[somk] Stanza (SOMobjects Kernel)	84
[somnm] Stanza (Naming Service)	84
[somras] Stanza (Error Log and Trace Facility)	85
[somsec] Stanza (Security Service)	87
[SOM_POSSOM] Stanza (Persistent Object Service)	88
Chapter 6. SOMobjects Coexistence Considerations	91
Multisystem Coexistence Considerations	91
Data Sets to Put on Shared Volumes	91
SOM Subsystem Considerations	91
SOMobjects Server Considerations	92
SOMobjects Configuration Utility (SOM@CFG) Considerations	92
RACF Considerations	92
Sysplex-specific Coexistence Considerations	92
Kind of Sysplex Required	92
Data Sets to Put on Shared Volumes	92
SOM Subsystem Considerations	93
SOMobjects Server Considerations	93
SOMobjects Configuration Utility (SOM@CFG) Considerations	94
RACF Considerations	94
Chapter 7. Preparing the OpenEdition Environment to Run SOMobjects	
Utilities	95
Creating Dummy Files for SOMobjects Functions	95
Setting Up the STEPLIB Concatenation for SOMobjects	97
Chapter 8. Initializing somOS::Server-based Application Servers	99
Chapter 9. Setting up the ISPF Interface to the SOMobjects Compiler	101
Setting Up How the Panels Will Be Invoked	101
Standalone Considerations	101
Modify the Skeletons for Background Compilation	101
Chapter 10. Configuring a DSOM Host	103
DSOM Host: Creating a JCL Procedure for the SOMobjects Naming Server	103
DSOM Host: Defining Security Identity and Authorizations for the SOMobjects Naming Server	103
DSOM Host: Registering the SOMobjects Naming Server with WLM	104
DSOM Host: Enabling WLM Server Management	105
DSOM Host: Running the SOMobjects Configuration Utility (SOM@CFG)	105
DSOM Host: Recycling the SOM Subsystem	106
DSOM Host: Capacity Planning for Running Many Distributed Clients	106
DSOM Host: Configuring and Running an Application Server	107
Index	109

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates.

Any reference to an IBM product, program or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, New York 10594
USA

Examples in This Book

The examples in this book are samples only, created by IBM Corporation. These sample examples are not part of any standard or IBM product and are provided to you solely for the purpose of assisting you in the development of your applications. The examples are provided "as is." IBM makes no warranties, express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose, regarding the function or performance of these examples. IBM shall not be liable for any damages arising out of your use of the sample examples, even if they have been advised of the possibility of such damages.

These sample examples can be freely distributed, copied, altered, and incorporated into other software, provided that it bears the above disclaimer intact.

Programming Interface Information

This publication is intended to help system programmers get OS/390 SOMobjects up and running. This book documents information that is Diagnosis, Modification or Tuning Information provided by OS/390 SOMobjects.

Attention: Do not use this Diagnosis, Modification or Tuning Information as a programming interface.

Trademarks

The following terms are trademarks for the IBM Corporation in the United States and/or other countries:

AD/Cycle

C++/MVS

C/370

IBM

OpenEdition

RACF

SOM

SOMobjects

About This Book

This book describes how to configure and administer OS/390 SOMobjects (simply called SOMobjects in the rest of the book).

Most of the book supports SOMobjects RTL (run-time library). Chapter 9, “Setting up the ISPF Interface to the SOMobjects Compiler” on page 101 is used to support SOMobjects ADE (application development environment).

Who Should Use This Book

This book is for the following audiences:

- The **SOMobjects administrator**, who can act as a system programmer (when setting up the SOMobjects environment), a WLM administrator (when configuring and administering SOMobjects with WLM), a REGIMPL administrator (when registering servers and classes in the implementation repository), and a security administrator (when configuring security for SOMobjects).
- **System programmers** who wish to run applications that are enabled for SOMobjects with the SOMobjects runtime feature.

This book assumes that you are an experienced systems programmer, and that you know how to:

- Run jobs on OS/390
- Write job control language (JCL)
- Use OpenEdition
- Use Time Sharing Option/Extended (TSO/E)
- Use restructured extended executor language (REXX)
- Use Transmission Control Protocol/Internet Protocol (TCP/IP).

How This Book Is Organized

This book is organized in the following way:

- Chapter 1, “Configuring Your SOMobjects Environment” on page 1 includes the following topics:
 - “Before You Begin” on page 1
 - “Getting Started with Non-Distributed SOMobjects” on page 1
 - “Getting Started with Distributed SOMobjects (DSOM)” on page 7
- Chapter 2, “Configuring and Administering SOMobjects with Workload Manager (WLM)” on page 15
- Chapter 3, “Registering Classes and Servers in the Implementation Repository” on page 53
- Chapter 4, “Configuring and Administering Security for SOMobjects” on page 67
- Chapter 5, “Setting up Configuration Files” on page 75

- Chapter 7, “Preparing the OpenEdition Environment to Run SOMobjects Utilities” on page 95
- Chapter 6, “SOMobjects Coexistence Considerations” on page 91
- Chapter 8, “Initializing somOS::Server-based Application Servers” on page 99
- Chapter 9, “Setting up the ISPF Interface to the SOMobjects Compiler” on page 101
- Chapter 10, “Configuring a DSOM Host” on page 103

Where to Find More Information

Where necessary, this book references information in other books, using shortened versions of the book title. For complete titles and order numbers of the books for all products that are part of OS/390, see *OS/390 Information Roadmap*, GC28-1727.

Related SOMobjects Publications

The OS/390 SOMobjects publications library includes the following:

- *OS/390 V2R4.0 SOMobjects: Getting Started*
- *OS/390 V2R4.0 SOMobjects Configuration and Administration Guide*
- *OS/390 V2R4.0 SOMobjects Programmer's Guide*
- *OS/390 V2R4.0 SOMobjects Object Services*
- *OS/390 V2R4.0 SOMobjects Programmer's Reference, Volume 1*
- *OS/390 V2R4.0 SOMobjects Programmer's Reference, Volume 2*
- *OS/390 V2R4.0 SOMobjects Programmer's Reference, Volume 3*
- *OS/390 V2R4.0 SOMobjects Messages, Codes, and Diagnosis*

OS/390 SOMobjects books are also available in softcopy on the OS/390 Collection (SK2T-6700) CD-ROM. IBM provides one copy of the CD-ROM automatically with the basic material for OS/390. You can order additional copies for a fee.

Summary of Changes

**Summary of Changes
for GC28-1851-01
OS/390 Version 2 Release 4**

The book contains information previously presented in GC28-1851-00, which supports OS/390 Version 1 Release 3.

This book includes terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Chapter 1. Configuring Your SOMobjects Environment

To support running SOMobjects applications on your system, you must configure and run the SOMobjects environment. This chapter describes getting started with both non-distributed and distributed SOMobjects (DSOM) environments.

Before You Begin

Before you can get started, you must take the following steps:

- ___ 1. Ensure that the SOMobjects-related post-install steps have been completed, as described in the *OS/390 Program Directory* shipped with your OS/390 order.
- ___ 2. Ensure that the following elements are configured and running on your system:
 - TCP/IP base
For details, see *TCP/IP for MVS: Customization and Administration Guide*.
 - OpenEdition System Services and, if you want to run SOMobjects utilities in the OpenEdition shell, OpenEdition Application Services.
For details, see *OS/390 OpenEdition Planning*.
 - To run secure SOMobjects applications, the Security Server (RACF) is required.
- ___ 3. To run distributed SOMobjects applications your system must be configured to run in WLM goal mode.

For details, see *OS/390 MVS Planning: Workload Management*.

Note: This section assumes that your installation installs and initializes SOMobjects in the recommended fashion. If your installation installs SOMobjects differently, you need to configure the SOMobjects differently. For example, you might use a subsystem name other than SOM. Or you might install the SOMobjects data sets with a high level qualifier other than SOMMVS.

This chapter covers the following topics:

- “Getting Started with Non-Distributed SOMobjects”
- “Getting Started with Distributed SOMobjects (DSOM)” on page 7

Getting Started with Non-Distributed SOMobjects

This section describes how to configure your system to support running non-distributed SOMobjects applications. You will perform the following steps:

- ___ 1. Specify OS/390 initialization settings for the SOM subsystem
- ___ 2. Create a JCL procedure for the SOM subsystem
- ___ 3. Define a security identity for the SOM subsystem
- ___ 4. Specify configuration file values for the SOM subsystem
- ___ 5. Start the SOM subsystem

- ___ 6. Validate your configuration by running a non-distributed application (IBM supplies a sample application).

Specifying OS/390 Initialization Settings for the SOM Subsystem

SOMObjects configuration requires several additions to your OS/390 initialization settings. The following steps are required:

- ___ 1. Define SOMObjects as an OS/390 subsystem. Do this by adding the following statement to your IEFSSNxx member:

```
SOM,GOSAMSSI OS/390 SOM subsystem
```

Hints & Tips

By specifying GOSAMSSI, the SOM subsystem will run under the primary subsystem (JESx) by default when started. Omit GOSAMSSI for the SOM subsystem to run under the master subsystem by default when started.

For further information about specifying OS/390 subsystems, see the description of IEFSSNxx in *OS/390 MVS Initialization and Tuning Reference*.

- ___ 2. Add the SOMObjects authorized libraries to your APF list. Do this by adding the following statement to your PROGxx member:

```
APF ADD
  DSNAME(SOMMVS.SGOSLDA)
  VOLUME(vvvvvv)
APF ADD
  DSNAME(SOMMVS.SGOSLNKL)
  VOLUME(vvvvvv)
```

Note: vvvvvv is the VOLSER on which the specified data sets reside.

For further information about adding libraries to APF, see the description of PROGxx in *OS/390 MVS Initialization and Tuning Reference*.

- ___ 3. Add the SOMObjects linklist library to the OS/390 linklist. Do this by adding SOMMVS.SGOSLNKL

to your LNKLSTxx or PROGxx parmlib member.

For further information about adding libraries to the linklist, see the description of LNKLSTxx, PROGxx, and dynamic linklist support in *OS/390 MVS Initialization and Tuning Reference*.

- ___ 4. Ensure that you have sufficient OS/390 resources allocated through your IEASYSxx member to accommodate the following SOMObjects requirements:

- 200K of extended CSA (ECSA)
- 500K of extended LPA (ELPA)
- One system LX

Note: When SOMObjects stops and restarts, it reuses its system LX.

- For each additional server, 128 bytes of ECSA.
- For each additional client, 96 bytes of ECSA.

- The following ASIDs:
 - One for the SOM subsystem
 - One for the SOMobjects security server
 - One for the SOMobjects naming server
 - One for each application server

Note: The SOM subsystem's ASID becomes non-reusable if the SOM subsystem ends. The ASID remains non-reusable until after a system reIPL.

To account for these conditions, you might want to increase the number of:

- System LXs specified on the NSYSLX keyword in IEASYSxx.
- Reserved, startable ASIDs specified on the RSVSTRT keyword in IEASYSxx, to allow SOMobjects to obtain an address space when it is started.
- Reserved, non-reusable ASIDs specified on the RSVNONR parameter in the IEASYSxx parmlib member, in the event that SOMobjects ends and cannot start again until after a reIPL.

For additional information about sizing CSA, specifying LXs, and reserving ASIDs, see *OS/390 MVS Initialization and Tuning Guide*.

5. Re-IPL OS/390 to make these settings take effect.

Hints & Tips

You can avoid re-IPLing immediately by issuing the following operator commands:

```
SETSSI  ADD,SUBNAME=SOM,INITRTN=GOSAMSSI
SETPROG APF,ADD,DSNAME=SOMMVS.SGOSLDA,VOLUME=vvvvvv
SETPROG APF,ADD,DSNAME=SOMMVS.SGOSLNKL,VOLUME=vvvvvv
SETPROG LNKLST,DEFINE,NAME=SOM,COPYFROM=IPL
SETPROG LNKLST,ADD,NAME=SOM,DSNAME=SOMMVS.SGOSLNKL,VOLUME=vvvvvv
SETPROG LNKLST,ACTIVATE,NAME=SOM
```

For more information on these commands, see *OS/390 MVS System Commands*.

Creating a JCL Procedure for the SOM Subsystem

The SOM subsystem runs as an MVS started task. It must be started with a JCL procedure. Create a JCL procedure for it by copying the sample procedure, SOMMVS.SGOSJCL(GOS1DSOM), to SYS1.PROCLIB as member SOM. This procedure requires that the following data sets be cataloged on your system:

- CEE.SCEERUN (LE run time library)
- SOMMVS.SGOSLDA (SOMobjects APF-authorized library)
- SOMMVS.SGOSLOAD (SOMobjects run time library)
- SOMMVS.SGOSPROF (SOMobjects configuration file)

Defining a Security Identity for the SOM Subsystem

The SOM subsystem requires a user identity for authorization purposes. To do this, take the following steps:

1. Assign a user ID to the SOM subsystem by issuing the following RACF commands. IBM recommends specifying SOM for the user ID, as shown in the USER parameter below. If SOM already exists as a user ID or group name on your system, you can specify another user ID, but you must adjust other definitions to match.

```
SETROPTS CLASSACT(STARTED)
SETROPTS RACLIST(STARTED)
SETROPTS GENERIC(STARTED)
RDEFINE STARTED SOM.* STDATA(USER(SOM) GROUP(SYS1))
SETROPTS RACLIST(STARTED) REFRESH
```

where SOM is the name of the cataloged procedure used for the SOM subsystem

Hints & Tips

For backup purposes, in case the STARTED class is inadvertently deactivated, IBM recommends also updating the RACF started procedures table (ICHRIN03 module). For more information, see “Associating User IDs with the SOM Subsystem and SOMObjects Servers” on page 69. For complete information on the STARTED class, see *OS/390 Security Server (RACF) Security Administrator's Guide*.

2. Authorize the user ID associated with the SOM subsystem to OpenEdition. For specific instructions, see *OS/390 OpenEdition Planning*.

Specifying Configuration File Values for the SOM Subsystem

The SOM subsystem requires a configuration file containing initialization values. The SOM procedure uses the sample configuration files, SOMMVS.SGOSPROF(GOENVH) and SOMMVS.SGOSPROF(GOENV1).

Note: SOMMVS.SGOSPROF(GOENVH) contains a specification of the HOSTNAME for your system; each SOM subsystem must have a unique HOSTNAME specification, so this file cannot be shared across systems. SOMMVS.SGOSPROF(GOENV1) contains specifications for most other SOMObjects environment variables; this file may be shared across systems.

1. To prevent your changes to SOMMVS.SGOSPROF(GOENVH) from being lost by installing SOMObjects service or upgrades, do the following:
 - a. Copy SOMMVS.SGOSPROF to SOMMVS.SOMENV.INI
 - b. Change SOMMVS.SGOSPROF to SOMMVS.SOMENV.INI in SYS1.PROCLIB(SOM)
2. Edit GOENVH to set the HOSTNAME keyword. This keyword appears in both the [somb] and [SOMD_TCPIP] stanzas. Both instances of the HOSTNAME keyword must be set to the same value. The SOM subsystem's configuration file should look like:

```
[somd]
HOSTNAME=mvs.xyzcorp.com
:
[SOMD_TCPIP]
HOSTNAME=mvs.xyzcorp.com
```

Where *mvs.xyzcorp.com* is the IP name of the system on which the SOM subsystem will run.

Hints & Tips

To assist in initial diagnosis, specify **SOMErrorLogDisplayMsgs=YES** in the [somras] stanza.

For a description of configuration files, see Chapter 5, “Setting up Configuration Files” on page 75.

Starting the SOM subsystem

Start the SOM subsystem by taking the following steps:

1. Before you start the SOM subsystem for the first time, you must activate the following RACF classes:

```
SETROPTS CLASSACT(CBIND)
SETROPTS CLASSACT(SERVER)
SETROPTS CLASSACT(SOMDOBJJS)
```

Note: If you define profiles in these classes, the classes have the following functions:

CBIND Controls client access to an application server.

SERVER Controls server access as a WLM application server.

SOMDOBJJS Controls client access to SOM classes and methods.

2. Issue the following operator command:

```
START SOM
```

When you see message

```
GOS009I SOM/MVS SOM INITIALIZATION COMPLETE.
```

the SOM subsystem is ready for use.

Hints & Tips

Issue **START SOM,SUB=JESx** to start the SOM subsystem under the primary subsystem. Issue **START SOM,SUB=MSTR** to start the SOM subsystem under the master subsystem. If you start the SOM subsystem under the master subsystem, refer to comments in **SOMMVS.SGOSJCL(GOS1DSOM)** for changes required to run under the master subsystem.

Validating Your Configuration by Running a Non-Distributed Application

To validate your configuration, run the non-distributed version of the car example. This example runs in a batch environment.

Follow the instructions in the sample job found in SOMMVS.SGOSJCL(GOS0AUTO).

Getting Started with Distributed SOMobjects (DSOM)

This section describes how to configure your system (the “install host”) to support running distributed SOMobjects applications. You will perform the following steps:

- ___ 1. Complete the configuration steps specified in “Getting Started with Non-Distributed SOMobjects” on page 1
- ___ 2. Create a JCL procedure for SOMobjects servers
- ___ 3. Define a security identity and authorizations for SOMobjects servers
- ___ 4. Define SOMobjects servers to WLM
- ___ 5. Enable WLM server management
- ___ 6. Run the SOMobjects configuration utility (SOM@CFG)
- ___ 7. Recycle the SOM subsystem

Creating a JCL Procedure for SOMobjects Servers

The distributed SOMobjects environment requires two server address spaces: one for a SOMobjects naming server and one for a SOMobjects security server. Both servers run as started tasks. They require a JCL procedure, and are started automatically by the SOM subsystem. Create a JCL procedure for them by copying sample procedure SOMMVS.SGOSJCL(GOSOSSVR) to SYS1.PROCLIB as member GOSOSSVR.

Defining a Security Identity and Authorizations for SOMobjects Servers

Assign the SOM user identity to the SOMobjects naming and security servers:

```
RDEFINE STARTED GOSOSSVR.* STDATA(USER(SOM) GROUP(SYS1))
SETROPTS RACLIST(STARTED) REFRESH
```

where GOSOSSVR is the name of the cataloged procedure used for the naming server and security server, and any other object services servers.

For information on using the RACF started procedures table to back up the STARTED class definitions, see “Associating User IDs with the SOM Subsystem and SOMobjects Servers” on page 69.

Registering SOMobjects Servers with WLM

If a requested server is not already running, it will automatically be started by WLM when required by a SOMobjects client program. All SOMobjects servers must be defined to WLM. Use the IWMARIN0 TSO command to establish these definitions:

- ___ 1. Issue TSO IWMARIN0
- ___ 2. Select option **2**. *Extract the definition from the WLM couple data set*
Selecting option **2** assumes you already have an active WLM definition in use on your system. (If this does not describe your situation, refer to Chapter 2, “Configuring and Administering SOMobjects with Workload Manager (WLM)” on page 15 for further information.)
- ___ 3. Select option **9** *Application Environments*

The following is the screen for the application environment setup for the naming server.

```
Application-Environment  Notes  Options  Help
-----
                          Create an Application Environment
Command ==> _____
Application Environment . . . NAMINGSERVER
Description . . . . . SOM Naming Server
Subsystem Type . . . . . SOM
Procedure Name . . . . . GOSOSSVR
Start Parameters . . . . . PARS='-a NAMINGSERVER'
Limit on starting server address spaces for a subsystem instance:
3  1. No limit
   2. Single address space per system
   3. Single address space per sysplex
```

The following is the screen for the application environment setup for the security server.

```
Application-Environment  Notes  Options  Help
-----
                          Create an Application Environment
Command ==> _____
Application Environment . . . SECURITYSERVER
Description . . . . . SOM Security Server
Subsystem Type . . . . . SOM
Procedure Name . . . . . GOSOSSVR
Start Parameters . . . . . PARS='-a SECURITYSERVER'
Limit on starting server address spaces for a subsystem instance:
3  1. No limit
   2. Single address space per system
   3. Single address space per sysplex
```

For detailed information, see Chapter 2, “Configuring and Administering SOMobjects with Workload Manager (WLM)” on page 15.

Enabling WLM Server Management

SOMobjects servers can be started only if WLM has been enabled to perform server management. Enable WLM server management by installing the definition to the couple data set, activating the service policy, and entering WLM goal mode.

Hints & Tips

Assuming a WLM policy named SOM30, use the following operator commands to activate the policy and enter goal mode:

```
VARY WLM,POLICY=SOM30
MODIFY WLM,MODE=GOAL
```

Note: IBM recommends that you set up performance goals for the servers. Unless performance goals are set up for the servers, the servers might receive a lower proportion of system resources than what might be expected.

See Chapter 2, “Configuring and Administering SOMobjects with Workload Manager (WLM)” on page 15 for information on performing these tasks.

Running the SOMobjects Configuration Utility (SOM@CFG)

The SOMobjects configuration utility (SOM@CFG) runs as a batch job that creates control files used by various SOMobjects facilities. These control files require 50 tracks of available DASD.

To run the SOMobjects configuration utility, take the following steps:

1. Ensure that the SOM@CFG job is submitted by a user ID that has the following authorizations:
 - Write access to SOMMVS.* data sets
 - OpenEdition access
2. Edit member GOSCFG in data set SOMMVS.SGOSJCL as follows:
3. Add job card information.
4. Submit the job.

Hints & Tips

Error Log Entries that can be Ignored

SOM@CFG creates error log entries that can be ignored if SOM@CFG completes successfully. You can expect log entries with the following error codes:

- Error code is 30046 [SOMDERROR_EntryNotFound]. SOM@CFG registers the Naming Service and Security Service in the Implementation Repository. When querying whether entries for these servers already exist, this error may be raised.
- Error code is 30088 [SOMDERROR_NamingNotActive]. When registering with the Implementation Repository, SOMobjects will attempt to store the information in the Naming Service. If the naming server is not running, this error will be raised.

Sysplex Considerations:

- Before running SOM@CFG in a sysplex, determine whether the VSAM clusters created by SOM@CFG need to be on DASD that is shared across the sysplex. For more information, see “Data Sets to Put on Shared Volumes” on page 91.
- SOM@CFG needs to be run only once for the entire sysplex, using the `-i` option (as specified in the IBM-supplied example).

Recycling the SOM subsystem

To complete the distributed configuration, the SOM subsystem must be re-cycled. Recycling the SOM subsystem is required so that it can reconfigure itself to incorporate data created by the SOMobjects configuration utility.

Re-cycle the SOM subsystem by issuing the following operator commands:

```
STOP SOM
START SOM
```

Note: The SOMobjects naming and security servers will terminate when the SOM subsystem is recycled. After the SOM subsystem is recycled, these servers will start on demand when client requests are initiated.

Configuring and Running an Application Server

To validate your configuration, you will run the distributed version of the car example. This example will run in a batch environment.

The distributed example consists of two components: 1) a client application (Driver); and 2) a server application (Car). Additional configuration is required to prepare the server application for running.

To configure the car server and run the Driver client, you will perform the following steps:

- ___ 1. Create a JCL procedure for the application server
- ___ 2. Define the security identity and authorization for the application server
- ___ 3. Define the application server to WLM
- ___ 4. Register the application server with the SOM subsystem
- ___ 5. Validate your configuration by running a distributed application (IBM supplies a sample application).

Hints & Tips

The configuration steps outlined in this section are applicable to any SOMobjects application server.

Creating a JCL Procedure for the Application Server

The application server (CARSERVER1, in this example) will run as a started task. This server will start automatically when its services are requested by the client application (Driver in this case).

- ___ 1. Create a JCL procedure for the application server by copying the sample procedure, SOMMVS.SGOSJCL(GOSSVR), to SYS1.PROCLIB as member GOSSVR.
- ___ 2. The classes that run under control of the Application Server reside in a DLL class library. This DLL class library is a partitioned data set (PDS) and must be allocated with the following attributes:
 - RECFM= U
 - LRECL= 0
 - BLKSIZE= 6144

This PDS must be added to the STEPLIB DD statement of your copy of the GOSSVR JCL procedure.

Note: The classes for CARSERVER1 will be link-edited by the job you are directed to run in "Validating Your Configuration by Running a Distributed Application" on page 13.

Defining Security Identity and Authorization for the Application Server

An application server requires a user identity for authorization purposes. To do this, take the following steps:

1. Assign a user identity to the GOSSVR procedure and authorize it to be the car server.

```
RDEFINE STARTED GOSSVR.* STDATA(USER(GOSSVR) GROUP(SYS1))
SETROPTS RACLIST(STARTED) REFRESH
RDEFINE SERVER (SOM.SOM.CARSERVER1) UACC(NONE)
PERMIT SOM.SOM.CARSERVER1 CLASS(SERVER) ID(GOSSVR) ACCESS(READ)
SETROPTS RACLIST(SERVER) REFRESH
```

2. Give user GOSSVR authorization to OpenEdition. For specific instructions, see *OS/390 OpenEdition Planning*.

Refer to Chapter 4, “Configuring and Administering Security for SOMobjects” on page 67 for additional information.

Defining the Application Server to WLM

Use IWMARIN0, as was done with the naming and security servers, to define the car application to WLM:

The following is the application environment setup for the car server.

```
Application-Environment  Notes  Options  Help
-----
                          Create an Application Environment
Command ==> _____
Application Environment . . . CARSERVER1
Description . . . . . Car Server
Subsystem Type . . . . . SOM
Procedure Name . . . . . GOSSVR
Start Parameters . . . . . PARMs='-a CARSERVER1'
Limit on starting server address spaces for a subsystem instance:
3  1. No limit
   2. Single address space per system
   3. Single address space per sysplex
```

For more information on defining your server application to WLM, see Chapter 2, “Configuring and Administering SOMobjects with Workload Manager (WLM)” on page 15.

Registering the Application Server with the SOM subsystem

Application servers must be registered with the SOM subsystem, so they can be located and accessed by client applications. Register the car server and the object class it supports, car, by issuing the following TSO commands:

```
REGIMPL POSIX(ON)/-A -i CARSERVER1
REGIMPL POSIX(ON)/-a -c Car -i CARSERVER1
REGIMPL POSIX(ON)/-a -c Dvehicle -i CARSERVER1
```

Notes:

1. SOMMVS.SGOSLOAD must be in your STEPLIB (or equivalent) for the REGIMPL utility to be invoked.
2. If you are using the object services server (somossvr), the database must be set up with the `somossvr -i` option. For more information, see Chapter 8, “Initializing somOS::Server-based Application Servers” on page 99.
3. For more information on registering your application servers and classes, see Chapter 3, “Registering Classes and Servers in the Implementation Repository” on page 53.

Hints & Tips

The instructions in this chapter configure an non-secure application server. This means clients may freely establish binds to this server, without authorization.

To set up secure servers, do the following:

1. Secure your application server by doing one of the following:
 - Specify `-s on` on the REGIMPL command as shown in the following:
`REGIMPL POSIX(ON)/-A -i CARSERVER1 -s on`
 - Change the server's SOMobjects registration by using the REGIMPL utility interactively. Choose option 3 (Change) and answer `y` to the prompt `Make server secure?.` For more information on changing a server's registration, see “Changing a Server in the Implementation Repository” on page 56.
2. You must authorize clients to bind to secure servers by setting up profiles in the RACF CBIND general resource class. For additional information on authorizing clients to servers, see Chapter 4, “Configuring and Administering Security for SOMobjects” on page 67.

Capacity Planning for Running Many Distributed Clients

As the number of clients being run increases, the number of socket connection requests to the daemon and servers which they access also increases. The rate at which the daemon and servers are able to accept client connect requests may decrease to the point where the backlog queue of socket connect requests waiting for an accept is exceeded. When this happens, the client receives a communication error exception resulting from a timeout received while attempting the connection request. To circumvent this, your installation might need to increase the SOMAXCONN configuration parameter in your TCPIP.PROFILE data set. The value should be increased to the greater of:

- The number of servers you anticipate will be connecting to the daemon simultaneously.
- The number of clients you anticipate will be connecting to your SOMobjects servers simultaneously.

You may also find as you increase the number of clients running against your servers that additional system tuning may be required. If unexpected exceptions (such as communication error exceptions) begin to occur you should also examine

| runtime diagnostic messages that may be displayed by programs such as TCP/IP.
| For example, TCP/IP may display error messages under the SYSDEBUG ddname.
| Also, if the SNALINK LU6.2 interface is configured with TCP/IP, the SYSPRINT
| ddname may contain error messages displayed by SNA.

Validating Your Configuration by Running a Distributed Application

To validate your configuration, run the distributed version of the car example. This example runs in a batch environment.

Follow the instructions in the sample job found in SOMMVS.SGOSJCL(GOS0DAUT).

Note: This job builds both the class DLLs that are required by the application server, as well as the client application. The class DLLs for the application server must be link-edited into the same PDS that was allocated and added to the application server's STEPLIB in "Creating a JCL Procedure for the Application Server" on page 10.

Chapter 2. Configuring and Administering SOMobjects with Workload Manager (WLM)

This chapter is intended for the workload manager (WLM) system administrator.

WLM manages workload distribution, workload balancing, and provides a solution for distributing resources to competing workloads.

The following sections describe how to define and manage servers using WLM:

- “Defining Servers to Workload Manager (WLM)”
- “Managing Performance with Workload Manager (WLM)” on page 30
- “Managing Servers Using WLM Commands” on page 51

Hints & Tips

If you are migrating from OS/390 Release 1, 2, or 3, make sure to see the migration information in *OS/390 MVS Planning: Workload Management*.

Defining Servers to Workload Manager (WLM)

The following describes how to configure the servers necessary for SOMobjects, as well as the additional configuration required to define the sample distributed application server called CARSERVER1.

- “Defining the Application Environment for the Naming Server” on page 17
- “Defining the Application Environment for the Security Server” on page 22
- “Defining an Application Server (CARSERVER1) to WLM” on page 26

We'll begin by invoking the main menu panel of WLM.

Invoke WLM by selecting the option for it on your ISPF panel or issue the following command from your command line:

```
TSO IWMARIN0
```

The following screen is displayed.

```

File Help
-----
Command ==> _____

      W  W  L    M  M
      W  W  L    MM MM
      W W W  L    M M M
      WW WW  L    M  M
      W  W  LLLL M  M

Licensed Materials - Property of IBM
5645-001 (C) Copyright IBM Corp. 1997.
All rights reserved.

ENTER to continue

```

Press **Enter**. The following screen is displayed.

```

File Help
-----
Command ==> _____

*****
*          Choose Service Definition          *
*                                          *
* Select one of the following options.      *
* 1. Read saved definition                  *
* 2. Extract definition from WLM           *
*    couple data set                       *
* 3. Create new definition                  *
*                                          *
* F1=Help      F2=Split      F5=KeysHelp   *
* F9=Swap      F12=Cancel    *
*****
ENTER to continue

```

Enter option **3** if you are creating a new definition. Enter option **1** or **2** if you already have a WLM definition defined and just need to update it. Press **Enter**.

```

File Help
-----
Command ==> _____

*****
*           Choose Service Definition           *
*                                               *
* Select one of the following options.         *
* 3_ 1. Read saved definition                   *
*    2. Extract definition from WLM             *
*       couple data set                       *
*    3. Create new definition                   *
*                                               *
* F1=Help      F2=Split      F5=KeysHelp      *
* F9=Swap      F12=Cancel                                          *
*****
ENTER to continue

```

The following Definition menu will be displayed.

```

File Utilities Notes Options Help
-----
WLM           Definition Menu           LEVEL003
Command ==> _____

Definition data set . . : none

Definition name . . . . . _____ (Required)
Description . . . . . _____

Select one of the
following options. . . . _ 1. Policies
                           2. Workloads
                           3. Resource Groups
                           4. Service Classes
                           5. Classification Groups
                           6. Classification Rules
                           7. Report Classes
                           8. Service Coefficients/Options
                           9. Application Environments
                          10. Scheduling Environments

```

Defining the Application Environment for the Naming Server

A naming server must be defined to WLM. To do so, perform the following tasks.

Select option **9** to add an application environment for the naming server.

```

File Utilities Notes Options Help
-----
WLM                               Definition Menu                               LEVEL003
Command ==> _____

Definition data set . . : none

Definition name . . . . . SOM_____ (Required)
Description . . . . . _____

Select one of the
following options. . . . 9_
                          1. Policies
                          2. Workloads
                          3. Resource Groups
                          4. Service Classes
                          5. Classification Groups
                          6. Classification Rules
                          7. Report Classes
                          8. Service Coefficients/Options
                          9. Application Environments
                          10. Scheduling Environments

```

Press **Enter**. The following screen is displayed.

```

Application-Environment Notes Options Help
-----
                                Create an Application Environment
Command ==> _____

Application Environment . . . _____ Required
Description . . . . . _____
Subsystem Type . . . . . _____ Required
Procedure Name . . . . . _____
Start Parameters . . . . . _____
                               _____
                               _____

Limit on starting server address spaces for a subsystem instance:
1  1. No limit
   2. Single address space per system
   3. Single address space per sysplex

*****
* Selection List empty. Define an application environment. (IWMAM600) *
*****

```

Fill in the panel and change the limit on starting server address spaces from a **1** to a **3** (Single address space per sysplex).

Application Environment: The alias of the naming server (NAMINGSERVER) as defined in NAMING_SERVER_ALIAS environment variable in the [somm] stanza.

Description: SOM naming server is the description we've chosen.

Subsystem Type: SOM (the name of the SOM subsystem).

Procedure name: GOSOSSVR (the name of the cataloged procedure that starts the naming server).

Start Parameters: The syntax to start the naming server is

PARMS='-a NAMINGSERVER'

Note: The alias specified in PARMS must match the application environment name and the alias given to the server with the REGIMPL utility. See “Adding a Server to the Implementation Repository” on page 54.

Exit after filling in these fields.

```
Application-Environment  Notes  Options  Help
-----
                          Create an Application Environment
Command ==> _____

Application Environment . . . NAMINGSERVER                      Required
Description . . . . . SOM naming server
Subsystem Type . . . . . SOM Required
Procedure Name . . . . . GOSOSSVR
Start Parameters . . . . . PARMS='-a NAMINGSERVER' _____
                                     _____
                                     _____

Limit on starting server address spaces for a subsystem instance:
3  1. No limit
   2. Single address space per system
   3. Single address space per sysplex
```

The following screen is displayed.

```
Application-Environment  Notes  Options  Help
-----
                          Application Environment Selection List      Row 1 to 1 of 1
Command ==> _____

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
              /=Menu Bar

Action  Application Environment Name      Description
___    NAMINGSERVER                      SOM naming server
***** Bottom of data *****

*****
* Application environment NAMINGSERVER was created. (IWMAM604) *
*****
```

Exit back to the Definition Menu.

```

File Utilities Notes Options Help
-----
WLM                               Definition Menu                               LEVEL003
Command ==> _____

Definition data set . . : none

Definition name . . . . . SOM          (Required)
Description . . . . . _____

Select one of the
following options. . . . _ 1. Policies
                           2. Workloads
                           3. Resource Groups
                           4. Service Classes
                           5. Classification Groups
                           6. Classification Rules
                           7. Report Classes
                           8. Service Coefficients/Options
                           9. Application Environments
                          10. Scheduling Environments

```

Click on **File**, and select option **4** to save the definition.

```

File Utilities Notes Options Help
*****
* 4 1. New          *          Definition Menu                               LEVEL003
* 2. Open          *          _____
* 3. Save          *
* 4. Save as      * : none
* 5. Print         *
* 6. Print as GML * . SOM          (Required)
* 7. Cancel       * . _____
* 8. Exit         *
*****
following options. . . . _ 1. Policies
                           2. Workloads
                           3. Resource Groups
                           4. Service Classes
                           5. Classification Groups
                           6. Classification Rules
                           7. Report Classes
                           8. Service Coefficients/Options
                           9. Application Environments
                          10. Scheduling Environments

```

Enter the name of a data set into which to save the definition. For this example we'll use **user1.som.wlm**. Press **Enter**.

```

File Utilities Notes Options Help
-----
WLM                               Definition Menu                               LEVEL003
Command ==> _____

Definition data set . . : none

Definition name . . . . . SOM          (Required)
Description . . . . . _____

Select one of the
following options. . . . _ 1. Policies
*****
*                               Save to...                               *
*                               *                                         *
*                               *                                         *
* Data set name . . . 'user1.som.wlm' _____ *
*                               *                                         *
*                               *                                         *
*                               *                                         *
*****

```

After getting the following screen, select option **1** to create the data set and press **Enter**.

```

File Utilities Notes Options Help
-----
WLM                               Definition Menu                               LEVEL003
Command ==> _____

Definition data set . . : none

Definition name . . . . . SOM          (Required)
Description . . . . . _____

Select one of the
following options. . . . _ 1. Policies
*****
*                               Save to...                               *
*                               *                                         *
*                               *                                         *
* Data set name . . . 'user1.som.wlm' _____ *
*                               *                                         *
*                               *                                         *
*                               *                                         *
*                               *                                         *
***** * Unable to find 'USER1.SOM.WLM' *
*                               *                                         *
* Do you want to create a new data set with this name? *
*                               *                                         *
* Select one of the following options. *
* 1_ 1. Create the data set. *
* 2. Do not create the data set. *
*                               *
*                               *
*                               *
*****

```

The following screen confirms that the user1.som.wlm service definition was saved.

```

File Utilities Notes Options Help
-----
WLM                               Definition Menu                LEVEL003
Command ==> _____

Definition data set . . : 'USER1.SOM.WLM'

Definition name . . . . . SOM          (Required)
Description . . . . . _____

Select one of the
following options. . . . _ 1. Policies
                           2. Workloads
                           3. Resource Groups
                           4. Service Classes
                           5. Classification Groups
                           6. Classification Rules
                           7. Report Classes
                           8. Service Coefficients/Options
                           9. Application Environments
                           10. Scheduling Environments

*****
* Service definition was saved. (IWMAM081) *
*****

```

Defining the Application Environment for the Security Server

The next step is to define the application environment for the security server. Your definition data set, USER1.SOM.WLM, will be used in this step.

Select option **9** to add a application environment for the security server and press **Enter**.

```

File Utilities Notes Options Help
-----
WLM                               Definition Menu                LEVEL003
Command ==> _____

Definition data set . . : 'USER1.SOM.WLM'

Definition name . . . . . SOM          (Required)
Description . . . . . _____

Select one of the
following options. . . . 9_ 1. Policies
                           2. Workloads
                           3. Resource Groups
                           4. Service Classes
                           5. Classification Groups
                           6. Classification Rules
                           7. Report Classes
                           8. Service Coefficients/Options
                           9. Application Environments
                           10. Scheduling Environments

```

After getting the following screen, enter action code **1** to create an application environment for the security server and press **Enter**.

```

Application-Environment  Notes  Options  Help
-----
Application Environment Selection List          Row 1 to 1 of 1
Command ==> _____

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
              /=Menu Bar

Action  Application Environment Name      Description
1_     NAMINGSERVER                      SOM naming server
***** Bottom of data *****

```

This results in the following panel.

```

Application-Environment  Notes  Options  Help
-----
Create an Application Environment
Command ==> _____

Application Environment . . . _____ Required
Description . . . . . _____
Subsystem Type . . . . . ____ Required
Procedure Name . . . . . _____
Start Parameters . . . . . _____
_____
_____

Limit on starting server address spaces for a subsystem instance:
1  1. No limit
   2. Single address space per system
   3. Single address space per sysplex

```

Fill in the panel and change the limit on starting server address spaces from a **1** to a **3**.

Application Environment: The alias of the security server (SECURITYSERVER) as defined in SECURITY_SERVER_ALIAS environment variable in the [somsec] stanza.

Description: SOM Security Server is the description we've chosen.

Subsystem Type: SOM (the name of the SOM subsystem).

Procedure name: GOSOSSVR (the name of the cataloged procedure that starts the naming server).

Start Parameters: The syntax to start the naming server is

```
PARMS='-a SECURITYSERVER'
```

Note: The alias specified in PARMS must match the application environment name and the alias given to the server with the REGIMPL utility. See "Adding a Server to the Implementation Repository" on page 54.

Exit after filling in these fields.

```

Application-Environment  Notes  Options  Help
-----
                                Create an Application Environment
Command ==> _____

Application Environment . . . SECURITYSERVER_____ Required
Description . . . . . SOM Security Server_____
Subsystem Type . . . . . som_ Required
Procedure Name . . . . . GOSOSSVR
Start Parameters . . . . . PARM='-a SECURITYSERVER' _____
                                _____
                                _____

Limit on starting server address spaces for a subsystem instance:
3  1. No limit
   2. Single address space per system
   3. Single address space per sysplex

```

The following screen is displayed.

```

Application-Environment  Notes  Options  Help
-----
                                Application Environment Selection List      Row 1 to 2 of 2
Command ==> _____

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
              /=Menu Bar

Action  Application Environment Name      Description
  _     NAMINGSERVER                      SOM naming server
  _     SECURITYSERVER                     SOM Security Server
***** Bottom of data *****

*****
* Application environment SECURITYSERVER was created. (IWAM604) *
*****

```

Exit back to the Definition Menu.

```

File Utilities Notes Options Help
-----
WLM                               Definition Menu           LEVEL003
Command ==> _____

Definition data set . . : 'USER1.SOM.WLM'

Definition name . . . . . SOM           (Required)
Description . . . . . _____

Select one of the
following options. . . . — 1. Policies
                             2. Workloads
                             3. Resource Groups
                             4. Service Classes
                             5. Classification Groups
                             6. Classification Rules
                             7. Report Classes
                             8. Service Coefficients/Options
                             9. Application Environments
                             10. Scheduling Environments

```

Select option **3** under **File** to save the definition and press **Enter**.

```

File Utilities Notes Options Help
*****
* 3 1. New           *      Definition Menu           LEVEL003
* 2. Open           *
* 3. Save           *
* 4. Save as        * : 'USER1.SOM.WLM'
* 5. Print          *
* 6. Print as GML * . SOM           (Required)
* 7. Cancel         * . _____
* 8. Exit           *
*****
following options. . . . — 1. Policies
                             2. Workloads
                             3. Resource Groups
                             4. Service Classes
                             5. Classification Groups
                             6. Classification Rules
                             7. Report Classes
                             8. Service Coefficients/Options
                             9. Application Environments
                             10. Scheduling Environments

```

The following screen confirms that the user1.som.wlm service definition was saved.

```

File Utilities Notes Options Help
-----
WLM                               Definition Menu                               LEVEL003
Command ==> _____

Definition data set . . . : 'USER1.SOM.WLM'

Definition name . . . . . SOM          (Required)
Description . . . . . _____

Select one of the
following options. . . . _ 1. Policies
                           2. Workloads
                           3. Resource Groups
                           4. Service Classes
                           5. Classification Groups
                           6. Classification Rules
                           7. Report Classes
                           8. Service Coefficients/Options
*****
* Service definition was saved. (IWMAM081) *
*****

```

Defining an Application Server (CARSERVER1) to WLM

We will now define the application server (CARSERVER1) to an application environment.

Select option **9** to define an application environment for CARSERVER1.

```

File Utilities Notes Options Help
-----
WLM                               Definition Menu                               LEVEL003
Command ==> _____

Definition data set . . . : 'USER1.SOM.WLM'

Definition name . . . . . SOM          (Required)
Description . . . . . _____

Select one of the
following options. . . . 9_ 1. Policies
                           2. Workloads
                           3. Resource Groups
                           4. Service Classes
                           5. Classification Groups
                           6. Classification Rules
                           7. Report Classes
                           8. Service Coefficients/Options
                           9. Application Environments
                          10. Scheduling Environments

```

Press **Enter**. The following screen is displayed.


```

Application-Environment  Notes  Options  Help
-----
Application Environment Selection List      Row 1 to 2 of 2
Command ==> _____

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
              /=Menu Bar

Action  Application Environment Name      Description
-----
  _     NAMINGSERVER                      SOM naming server
  _     SECURITYSERVER                    SOM Security Server
***** Bottom of data *****

```

Enter action code **1** to create an application environment for the car server and press **Enter**.

```

Application-Environment  Notes  Options  Help
-----
Application Environment Selection List      Row 1 to 2 of 2
Command ==> _____

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
              /=Menu Bar

Action  Application Environment Name      Description
-----
  1     NAMINGSERVER                      SOM naming server
  1     SECURITYSERVER                    SOM Security Server
***** Bottom of data *****

```

This results in the following panel.

```

Application-Environment  Notes  Options  Help
-----
Create an Application Environment
Command ==> _____

Application Environment . . . _____ Require
Description . . . . . _____
Subsystem Type . . . . . _____ Required
Procedure Name . . . . . _____
Start Parameters . . . . . _____
                          _____
                          _____

Limit on starting server address spaces for a subsystem instance:
1  1. No limit
   2. Single address space per system
   3. Single address space per sysplex

```

Fill in the panel and change the limit on starting server address spaces from a **1** to a **3**.

Application Environment: The alias assigned to the server with the REGIMPL utility. For this example, CARSERVER1.

Description: SOM Car Server is the description we've chosen.

Subsystem Type: SOM (the name of the SOM subsystem).

Procedure name: GOSSVR (the name of the cataloged procedure that starts the generic server program).

Note: GOSSVR needs to be copied from SOMMVS.SGOSJCL(GOSSVR) to SYS1.PROCLIB. If you update the GOSSVR procedure to use a configuration file on the SOMENV DD statement, make sure that HOSTNAME is not specified in the configuration file specified. If HOSTNAME is specified, then the procedure will only start successfully on one system in the sysplex.

Start Parameters: The syntax to start the naming server is

PARMS='-a CARSERVER1'.

Note: The alias specified in PARMS must match the application environment name and the alias given to the server with the REGIMPL utility. See "Adding a Server to the Implementation Repository" on page 54.

Exit.

```

Application-Environment  Notes  Options  Help
-----
                          Create an Application Environment
Command ==> _____

Application Environment . . . CARSERVER1_____ Require
Description . . . . . SOM Car Server_____
Subsystem Type . . . . . SOM_ Required_____
Procedure Name . . . . . GOSSVR_____
Start Parameters . . . . . PARMS='-a CARSERVER1'_____

Limit on starting server address spaces for a subsystem instance:
3  1. No limit
   2. Single address space per system
   3. Single address space per sysplex

```

The following screen is displayed.

```

Application-Environment  Notes  Options  Help
-----
                          Application Environment Selection List      Row 1 to 3 of 3
Command ==> _____

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
              /=Menu Bar

Action  Application Environment Name      Description
-----  -
   1    CARSERVER1                       SOM Car Server
   2    NAMINGSERVER                     SOM naming server
   3    SECURITYSERVER                   SOM Security Server
***** Bottom of data *****

*****
* Application environment CARSERVER1 was created. (IWMAM604) *
*****

```

Exit back to the Definition Menu.

```

File Utilities Notes Options Help
-----
WLM                               Definition Menu                               LEVEL003
Command ==> _____

Definition data set . . . : 'USER1.SOM.WLM'

Definition name . . . . . SOM           (Required)
Description . . . . . _____

Select one of the
following options. . . . _ 1. Policies
                             2. Workloads
                             3. Resource Groups
                             4. Service Classes
                             5. Classification Groups
                             6. Classification Rules
                             7. Report Classes
                             8. Service Coefficients/Options
                             9. Application Environments
                             10. Scheduling Environments

```

Select option **3** under **File** to save the definition and press **Enter**.

```

File Utilities Notes Options Help
*****
* 3 1. New          *           Definition Menu                               LEVEL003
* 2. Open          * _____
* 3. Save          *
* 4. Save as      * : 'USER1.SOM.WLM'
* 5. Print        *
* 6. Print as GML * . SOM           (Required)
* 7. Cancel       * . _____
* 8. Exit         *
*****
following options. . . . _ 1. Policies
                             2. Workloads
                             3. Resource Groups
                             4. Service Classes
                             5. Classification Groups
                             6. Classification Rules
                             7. Report Classes
                             8. Service Coefficients/Options
                             9. Application Environments
                             10. Scheduling Environments

```

The following screen confirms that the user1.som.wlm service definition was saved.

Note: You could have created the NAMINGSERVER, SECURITYSERVER and CARSERVER1 all at one time, and then saved the service definition just once.

```

File Utilities Notes Options Help
-----
WLM                               Definition Menu                               LEVEL003
Command ==> _____

Definition data set . . : 'USER1.SOM.WLM'

Definition name . . . . . SOM          (Required)
Description . . . . . _____

Select one of the
following options. . . . _ 1. Policies
                           2. Workloads
                           3. Resource Groups
                           4. Service Classes
                           5. Classification Groups
save                        6. Classification Rules
                           7. Report Classes
                           8. Service Coefficients/Options
*****
* Service definition was saved. (IWMAM081) *
*****

```

Managing Performance with Workload Manager (WLM)

OS/390 provides the ability to control resources that are expended on workloads to enable them to meet installation goals. This function is provided by the OS/390 Workload Manager (WLM). WLM manages the system resources, such as CPU cycles and memory, that are assigned to a unit of work to enable it to meet goals set forth by the installation.

WLM manages a set of like work that is referred to as a service class. Service classes are assigned a set of goals that WLM will work towards meeting.

As a distributed request enters the system, the SOM subsystem uses WLM to assign a service class to the request. In SOMobjects, a unit of work is defined as a single remote method call.

Using WLM, you can choose among four types of rules for setting up service classes to handle SOMobjects work:

1. A default service class for all SOMobjects work that is not assigned to a class via another classification rule.
2. A service class based upon the server address space in which the work is running. This is referred to as a collection name. When defining this rule to WLM, the application environment name is used as the collection name.

Hints & Tips

IBM recommends setting higher priority goals for the naming and security servers than for other application servers.

3. A service class based upon the client that submitted the work.
4. A service class based upon the SOMobjects class, and the method being invoked.

The SOM subsystem, naming server, security server and any other application servers should be defined to use a service class or classes that enable them to attain the desired performance goals.

Following is a set of some definitions for the Carserver, divided into the following topics:

- “Creating a Service Class” on page 32
- “Creating Classification Rules” on page 36
- “Creating a Service Policy” on page 44
- “Installing a Definition” on page 47
- “Activating the Service Policy” on page 48

Creating a Workload

The first step is to define a workload. A workload is a named collection of work to be tracked and reported as a unit. You arrange workloads by subsystem (SOM) or by major applications (production, batch, office) or by line of business (ATM, inventory department). Logically, a workload is a set of service classes.

A workload needs to be created in WLM in order to install the definition. The following tasks will walk you through creating a workload.

Select option **2** to create a workload.

```
File Utilities Notes Options Help
-----
WLM                               Definition Menu                LEVEL003
Command ==> _____

Definition data set . . . : 'USER1.SOM.WLM'

Definition name . . . . . SOM          (Required)
Description . . . . . _____

Select one of the
following options. . . . 2  1. Policies
                           2. Workloads
                           3. Resource Groups
                           4. Service Classes
                           5. Classification Groups
                           6. Classification Rules
                           7. Report Classes
                           8. Service Coefficients/Options
                           9. Application Environments
                          10. Scheduling Environments
```

The following screen is displayed.

```

Workload Notes Options Help
-----
                          Create a Workload
Command ==> _____

Enter or change the following information:

Workload Name . . . . . _____ (Required)
Description . . . . . _____

*****
* Selection List empty. Define a workload. (IWMAM200) *
*****

```

Enter workload name (we'll choose **somwork**) and then **Exit**.

```

Workload Notes Options Help
-----
                          Create a Workload
Command ==> _____

Enter or change the following information:

Workload Name . . . . . somwork_ (Required)
Description . . . . . _____

```

After the workload is created (see following screen), **Exit** back to the Definition Menu.

```

Workload View Notes Options Help
-----
                          Workload Selection List                               Row 1 to 1 of 1
Command ==> _____

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
              /=Menu Bar

Action Name      Description                               User      Date
-----
_  SOMWORK
***** Bottom of data *****

*****
* Workload SOMWORK was created. (IWMAM211) *
*****

```

Creating a Service Class

A service class is a named group of work within a workload with similar performance characteristics. You can create service classes for work with the same:

- Performance goals
- Resource requirements
- Business importance to the installation.

WLM manages a service class as a single entity when allocating resources to meet performance goals. You can assign the following kinds of performance goals to service classes:

- average response time
- response time with percentile
- velocity
- discretionary.

You assign an importance level to the performance goal. Importance indicates how vital it is to the installation that the performance goal be met relative to other goals.

Because some work has variable resource requirements, WLM provides performance periods where you can specify a series of varying goals and importance. You can define up to eight performance periods for each service class.

For workload management, you assign a service goal and optionally a resource group to a service class. A service class needs to be created in WLM in order to install the definition.

The following tasks will walk you through creating a service class.

Select option **4** to create a service class.

```
File Utilities Notes Options Help
-----
WLM                               Definition Menu
Command ==> _____

Definition data set . . : 'USER1.SOM.WLM'

Definition name . . . . . SOM          (Required)
Description . . . . . _____

Select one of the
following options. . . . 4_  1. Policies
                             2. Workloads
                             3. Resource Groups
                             4. Service Classes
                             5. Classification Groups
                             6. Classification Rules
                             7. Report Classes
                             8. Service Coefficients/Options
                             9. Application Environments
                             10. Scheduling Environments
```

The following screen is displayed.

```

Service-Class  Notes  Options  Help
-----
Create a Service Class                               Row 1 to
Command ==> _____

Service Class Name . . . . . _____ (Required)
Description . . . . . _____
Workload Name . . . . . _____ (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
-----
***** Bottom of data *****

*****
* Selection List empty. Define a service class. (IWMAM300) *
*****

```

Enter a name for the service class (we'll choose **SOMSERV**) and the name of the workload (**SOMWORK**) that was created previously.

Enter action code **I** to insert a new period and press **Enter**.

```

Service-Class  Notes  Options  Help
-----
Create a Service Class                               Row 1 to
Command ==> _____

Service Class Name . . . . . SOMSERV (Required)
Description . . . . . _____
Workload Name . . . . . SOMWORK (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
-----
  i_
***** Bottom of data *****

```

The following screen is displayed. Select the type of goal and press **Enter**. This example selects option **2**, response time with percentile.


```

Service-Class Notes Options Help
- *****
* Choose a goal type for period 1 * ss Row 1 to
C * _____
*
S * 2_ 1. Average response time * ired)
D * 2. Response time with percentile * _____
W * 3. Execution velocity * or ?)
B * 4. Discretionary * or ?)
*
S * F1=Help F2=Split F5=KeysHelp * I=Insert new period,
E * F9=Swap F12=Cancel *
*****
---Period--- -----Goal-----
Action # Duration Imp. Description
i
***** Bottom of data *****

```

Enter the desired percentile and response time. For illustration purposes, we'll choose **90** for percentile, and **10** for response time.

```

Service-Class Notes Options Help
- *****
* Choose a goal type for period 1 * ss Row 1 to
C * _____
*
S * 2_ 1. Average response time * ired)
D *****
W * Response time with percentile goal *
B *
* Enter a percentile and response time goal for period 1 *
S *
E * Percentile . . . 90 (1-99) *
*
* Hours . . . . . _ (0-24) *
A * Minutes . . . . _ (0-99) *
* Seconds . . . . 10 _ (0-9999) *
*
* Importance . . . 1 (1=highest, 5=lowest) *
* Duration . . . _____ (1-999,999,999, or *
* none for last period) *
*****

```

Save the changes by entering **Exit**.

```

Service-Class Notes Options Help
-----
                                Create a Service Class                Row 1 to
Command ==> _____

Service Class Name . . . . . SOMSERV   (Required)
Description . . . . . _____
Workload Name . . . . . SOMWORK   (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          1    90% complete within 00:00:10.000
***** Bottom of data *****

*****
* Press EXIT to save your changes or CANCEL to discard them. (IWMAM600) *
*****

```

The following screen is displayed. **Exit** to return to the Definition Menu, which follows.

```

Service-Class View Notes Options Help
-----
                                Service Class Selection List          Row 1 to 1 of 1
Command ==> _____

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
              /=Menu Bar

Action Class   Description                               Workload
  1    SOMSERV                               SOMWORK
***** Bottom of data *****

*****
* Service class SOMSERV was created. (IWMAM311) *
*****

```

Creating Classification Rules

Classification rules are the rules you define to relate your service class goals to incoming work. In the SOMobjects case, incoming units of work are method invocations inside the SOM server.

Classification rules categorize work into service classes based on work qualifiers. A work qualifier is what identifies a work request to the system, such as the

- name of the server that it is running in
- user ID
- class and method name of the request.

As described earlier there are four ways of assigning a service class to an incoming SOMobjects unit of work.

1. Allow the default classification to take place
2. Assign a service class based upon the server where it is running
3. Assign a service class based upon the client user ID
4. Assign a service class based upon the class and method names being invoked.

In order to do this, SOMobjects provides WLM this information for every method run in a SOMobjects server. This information is brought to the classification rules in the following ways.

- The server where the unit of work is running is provided as a collection name
- The client user ID passes as the user ID the work unit is running under
- The class and method names are provided in the subsystem-specific parameter (SPM). The first 123 bytes of the class name are provided in the first 123 bytes of the SPM, the first 123 bytes of the method name are provided in the next 123 bytes of the SPM.

There are standard ways that WLM creates classification rules, including selecting class and method names from the SPM. See *OS/390 MVS Workload Management Services* for how these rules are defined.

The following tasks will walk you through an example of each of the following classification rules:

- “Classifying Work by User ID” on page 38
- “Classifying Work by Class Name” on page 39
- “Classifying Work by Method Name” on page 40
- “Classifying Work by Application Environment Name” on page 41

To create classification rules, select option **6** and press **Enter**.

```

File Utilities Notes Options Help
-----
WLM                               Definition Menu
Command ==> _____

Definition data set . . . : 'USER1.SOM.WLM'

Definition name . . . . . SOM          (Required)
Description . . . . . _____

Select one of the
following options. . . . 6_  1. Policies
                             2. Workloads
                             3. Resource Groups
                             4. Service Classes
                             5. Classification Groups
                             6. Classification Rules
                             7. Report Classes
                             8. Service Coefficients/Options
                             9. Application Environments
                             10. Scheduling Environments

```

For SOMobjects, enter action code **3** to modify the SOM type and press **Enter**.

```

Subsystem-Type View Notes Options Help
-----
Subsystem Type Selection List for Rules      Row 1 to 11 of 20
Command ==> _____

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
              /=Menu Bar

Action  Type      Description                                -----Class-----
      Type      Description                                Service  Report
-----
   _   ASCH      Use Modify to enter YOUR rules
   _   CICS      Use Modify to enter YOUR rules
   _   DB2       Use Modify to enter YOUR rules
   _   DDF       Use Modify to enter YOUR rules
   _   IMS       Use Modify to enter YOUR rules
   _   IWEB     Use Modify to enter YOUR rules
   _   JES       Use Modify to enter YOUR rules
   _   LSFM     Use Modify to enter YOUR rules
   _   OMVS     Use Modify to enter YOUR rules
   3_   SOM      Use Modify to enter YOUR rules
   _   STC      Use Modify to enter YOUR rules

```

The following screen is displayed.

```

Subsystem-Type Xref Notes Options Help
-----
Modify Rules for the Subsystem Type          Row 1 to 1 of 1
Command ==> _____ SCROLL ==> PAGE

Subsystem Type . : SOM      Fold qualifier names?  Y (Y or N)
Description . . . Use Modify to enter YOUR rules

Action codes:  A=After      C=Copy          M=Move          I=Insert rule
                B=Before    D=Delete row    R=Repeat        IS=Insert Sub-rule
                -----Qualifier-----
Action  Type      Name      Start      Service  Report
-----
   _   1          _____
***** BOTTOM OF DATA *****

```

Enter service class **SOMSERV**, which was created previously as the default service class. Other service classes should be created and used for specific rules.

The rules defined in the following sections will all run under the default SOMSERV service class since a specific service class was not specified.

Classifying Work by User ID

Enter **UI** (to classify by user ID) in the Qualifier Type field and **USER1** in the Qualifier Name field. Press **Enter**.

```

Subsystem-Type Xref Notes Options Help
-----
Command ==> _____ Modify Rules for the Subsystem Type Row 1 to 1 of 1
SCROLL ==> PAGE

Subsystem Type . . : SOM          Fold qualifier names?  Y (Y or N)
Description . . . _____

Action codes:  A=After   C=Copy       M=Move       I=Insert rule
                B=Before   D=Delete row R=Repeat     IS=Insert Sub-rule
                -----Qualifier-----
Action   Type      Name      Start          Service      Report
_____ 1 UI      USER1    _____    _____    _____
***** BOTTOM OF DATA *****

```

Classifying Work by Class Name

Enter action code **I** to insert a subsystem parameter rule, and press **Enter**.

```

Subsystem-Type Xref Notes Options Help
-----
Command ==> _____ Modify Rules for the Subsystem Type Row 1 to 1 of 1
SCROLL ==> PAGE

Subsystem Type . . : SOM          Fold qualifier names?  Y (Y or N)
Description . . . _____

Action codes:  A=After   C=Copy       M=Move       I=Insert rule
                B=Before   D=Delete row R=Repeat     IS=Insert Sub-rule
                -----Qualifier-----
Action   Type      Name      Start          Service      Report
_____ 1 UI      USER1    _____    _____    _____
i_____ 1 UI      USER1    _____    _____    _____
***** BOTTOM OF DATA *****

```

The following screen is displayed.

```

Subsystem-Type Xref Notes Options Help
-----
Command ==> _____ Modify Rules for the Subsystem Type Row 1 to 2 of 2
SCROLL ==> PAGE

Subsystem Type . . : SOM          Fold qualifier names?  Y (Y or N)
Description . . . _____

Action codes:  A=After   C=Copy       M=Move       I=Insert rule
                B=Before   D=Delete row R=Repeat     IS=Insert Sub-rule
                -----Qualifier-----
Action   Type      Name      Start          Service      Report
_____ 1 UI      USER1    _____    _____    _____
_____ 1 _____    _____    _____    _____
***** BOTTOM OF DATA *****

```

Enter **SPM** (to classify by class name) in the Qualifier Type field. and the class name **DVEHICLE** in the Qualifier Name field.

```

Subsystem-Type Xref Notes Options Help
-----
Command ==> _____ Modify Rules for the Subsystem Type Row 1 to 2 of 2
SCROLL ==> PAGE

Subsystem Type . : SOM Fold qualifier names? Y (Y or N)
Description . . . _____

Action codes: A=After C=Copy M=Move I=Insert rule
              B=Before D=Delete row R=Repeat IS=Insert Sub-rule
              -----Qualifier-----
Action Type Name Start Service Report
              -----Class-----
              DEFAULTS: SOMSERV
_____ 1 UI USER1 _____
_____ 1 SPM DVEHICLE 1 _____
***** BOTTOM OF DATA *****

```

Classifying Work by Method Name

Enter Action code of **IS** in the Action field to create a sub-rule for method name.
Press **Enter**.

```

Subsystem-Type Xref Notes Options Help
-----
Command ==> _____ Modify Rules for the Subsystem Type Row 1 to 2 of 2
SCROLL ==> PAGE

Subsystem Type . : SOM Fold qualifier names? Y (Y or N)
Description . . . _____

Action codes: A=After C=Copy M=Move I=Insert rule
              B=Before D=Delete row R=Repeat IS=Insert Sub-rule
              -----Qualifier-----
Action Type Name Start Service Report
              -----Class-----
              DEFAULTS: SOMSERV
_____ 1 UI USER1 _____
is_ 1 SPM DVEHICLE 1 _____
***** BOTTOM OF DATA *****

```

The following screen is displayed.

```

Subsystem-Type Xref Notes Options Help
-----
Command ==> _____ Modify Rules for the Subsystem Type Row 1 to 3 of 3
SCROLL ==> PAGE

Subsystem Type . : SOM Fold qualifier names? Y (Y or N)
Description . . . _____

Action codes: A=After C=Copy M=Move I=Insert rule
              B=Before D=Delete row R=Repeat IS=Insert Sub-rule
              -----Qualifier-----
Action Type Name Start Service Report
              -----Class-----
              DEFAULTS: SOMSERV
_____ 1 UI USER1 _____
_____ 1 SPM DVEHICLE 1 _____
_____ 2 _____
***** BOTTOM OF DATA *****

```



```

Subsystem-Type Xref Notes Options Help
-----
                Modify Rules for the Subsystem Type          Row 1 to 4 of 4
Command ==> _____ SCROLL ==> PAGE

Subsystem Type . : SOM          Fold qualifier names?  Y (Y or N)
Description . . . _____

Action codes:  A=After   C=Copy       M=Move       I=Insert rule
                B=Before  D=Delete row  R=Repeat     IS=Insert Sub-rule
                -----Qualifier-----
Action   Type      Name      Start      Service      Report
                -----Class-----
                DEFAULTS: SOMSERV
_____  1  UI        USER1      _____
_____  1  SPM       DVEHICLE 1  _____
_____  2  SPM       START      124        _____
_____  1  _____ _____
***** BOTTOM OF DATA *****

```

Enter **CN** (which represents collection name) in the type files, and enter the first 8 characters of the application environment name (**CARSERVE**). Press **Enter**.

Note: Since these panels only allow 8 characters in the field, we will have to enter the rest of the CARSERVER1 name in following screens.

```

Subsystem-Type Xref Notes Options Help
-----
                Modify Rules for the Subsystem Type          Row 1 to 4 of 4
Command ==> _____ SCROLL ==> PAGE

Subsystem Type . : SOM          Fold qualifier names?  Y (Y or N)
Description . . . _____

Action codes:  A=After   C=Copy       M=Move       I=Insert rule
                B=Before  D=Delete row  R=Repeat     IS=Insert Sub-rule
                -----Qualifier-----
Action   Type      Name      Start      Service      Report
                -----Class-----
                DEFAULTS: SOMSERV
_____  1  UI        USER1      _____
_____  1  SPM       DVEHICLE 1  _____
_____  2  SPM       START      124        _____
_____  1  CN        CARSERVE    _____
***** BOTTOM OF DATA *****

```

Insert a sub-rule, **IS** (Insert), to enter the rest of the application environment name.


```

Subsystem-Type Xref Notes Options Help
-----
                Modify Rules for the Subsystem Type          Row 1 to 4 of 4
Command ==> _____ SCROLL ==> PAGE

Subsystem Type . . : SOM          Fold qualifier names?  Y (Y or N)
Description . . . _____

Action codes:  A=After   C=Copy       M=Move       I=Insert rule
                B=Before   D=Delete row  R=Repeat     IS=Insert Sub-rule
                -----Qualifier-----
Action   Type      Name      Start          Service      Report
                -----Class-----
                DEFAULTS: SOMSERV
_____  1  UI      USER1          _____
_____  1  SPM      DVEHICLE 1     _____
_____  2  SPM      START          124          _____
is_____ 1  CN      CARSERVE 1     _____
***** BOTTOM OF DATA *****

```

The following screen is displayed.

```

Subsystem-Type Xref Notes Options Help
-----
                Modify Rules for the Subsystem Type          Row 1 to 5 of 5
Command ==> _____ SCROLL ==> PAGE

Subsystem Type . . : SOM          Fold qualifier names?  Y (Y or N)
Description . . . _____

Action codes:  A=After   C=Copy       M=Move       I=Insert rule
                B=Before   D=Delete row  R=Repeat     IS=Insert Sub-rule
                -----Qualifier-----
Action   Type      Name      Start          Service      Report
                -----Class-----
                DEFAULTS: SOMSERV
_____  1  UI      USER1          _____
_____  1  SPM      DVEHICLE 1     _____
_____  2  SPM      START          124          _____
_____  1  CN      CARSERVE 1     _____
_____  2  _____          _____
***** BOTTOM OF DATA *****

```

Enter the rest of the application environment name, **R1** (it starts at position 9), and press **Enter**.

```

Subsystem-Type Xref Notes Options Help
-----
Command ==> _____ Modify Rules for the Subsystem Type Row 1 to 5 of 5
                          SCROLL ==> PAGE

Subsystem Type . . : SOM          Fold qualifier names? Y (Y or N)
Description . . . _____

Action codes: A=After C=Copy M=Move I=Insert rule
              B=Before D=Delete row R=Repeat IS=Insert Sub-rule
              -----Qualifier-----
Action  Type      Name      Start      Service Report
-----  -
_____ 1  UI         USER1
_____ 1  SPM        DVEHICLE 1
_____ 2  SPM        START      124
_____ 1  CN         CARSERVE 1
_____ 2  CN         R1         9
DEFAULTS: SOMSERV
***** BOTTOM OF DATA *****

```

Exit to return to the next screen"

```

Subsystem-Type View Notes Options Help
-----
Command ==> _____ Subsystem Type Selection List for Rules Row 1 to 11 of 20

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
              /=Menu Bar

Action  Type      Description      -----Class-----
-----  -
_____ ASCH      Use Modify to enter YOUR rules      Service Report
_____ CICS      Use Modify to enter YOUR rules
_____ DB2       Use Modify to enter YOUR rules
_____ DDF       Use Modify to enter YOUR rules
_____ IMS      Use Modify to enter YOUR rules
_____ IWEB     Use Modify to enter YOUR rules
_____ JES      Use Modify to enter YOUR rules
_____ LSFM     Use Modify to enter YOUR rules
_____ OMVS     Use Modify to enter YOUR rules
_____ SOM      *****
_____ STC      * Subsystem type SOM was changed. (IWMAM731) *
              *****

```

Exit to return to the Definition Menu, which is the first screen in "Creating a Service Policy."

At this point, each of the classification rules that have been saved will be managed to the same set of performance goals that were entered in the service class panels, since they are all part of that service class.

Creating a Service Policy

A service policy is a named set of performance goals that workload management uses as a guideline to match resources to work. A definition data set must have at least one service policy in order to install the definition and activate the service policy. Any one service policy can have one or more workloads defined in it. The following example shows only one workload, but you can define more than one workload as needed by your system.

The following tasks will walk you through creating a service policy.

Enter option **1** in the Definition Menu to create a new policy and press **Enter**.

```
File Utilities Notes Options Help
-----
WLM                               Definition Menu
Command ==> _____

Definition data set . . . : 'USER1.SOM.WLM'

Definition name . . . . . SOM      (Required)
Description . . . . . _____

Select one of the
following options. . . . 1_  1. Policies
                             2. Workloads
                             3. Resource Groups
                             4. Service Classes
                             5. Classification Groups
                             6. Classification Rules
                             7. Report Classes
                             8. Service Coefficients/Options
                             9. Application Environments
                             10. Scheduling Environments
```

The following screen is displayed.

```
Service-Policy Notes Options Help
-----
                               Create a Service Policy
Command ==> _____

Enter or change the following information:

Service Policy Name . . . . . _____ (Required)
Description . . . . . _____

*****
* Selection List empty. Define a service policy. (IWMAM100)*
*****
```

Select a name for the policy (we'll choose **SOM30**) and press **Enter**.

```
Service-Policy Notes Options Help                               Window mov
-----
                               Create a Service Policy
Command ==> _____

Enter or change the following information:

Service Policy Name . . . . . som30__ (Required)
Description . . . . . _____

*****
* Selection List empty. Define a service policy. (IWMAM100)*
*****
```

The following screen is displayed.

```
Service-Policy Notes Options Help
-----
                          Create a Service Policy
Command ==> _____

Enter or change the following information:
Service Policy Name . . . . . SOM30      (Required)
Description . . . . . _____

*****
* Press EXIT to save your changes or CANCEL to discard them. (IWMAM600) *
*****
```

Exit to return to the next screen.

```
Service-Policy View Notes Options Help
-----
                          Service Policy Selection List                      Row 1 to 1 of 1
Command ==> _____

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
              7=Override Service Classes, 8=Override Resource Groups,
              /=Menu Bar

Action Name      Description                                ----Last Change-----
   _   SOM30                                           User      Date
***** Bottom of data *****
              USER1  1996/10/21

*****
* Service policy SOM30 was created. (IWMAM111) *
*****
```

After the policy is created, **Exit** back to the Definition Menu.

```
File Utilities Notes Options Help
-----
WLM                          Definition Menu
Command ==> _____

Definition data set . . . : 'USER1.SOM.WLM'

Definition name . . . . . SOM      (Required)
Description . . . . . _____

Select one of the
following options. . . . — 1. Policies
                           2. Workloads
                           3. Resource Groups
                           4. Service Classes
                           5. Classification Groups
                           6. Classification Rules
                           7. Report Classes
                           8. Service Coefficients/Options
                           9. Application Environments
                          10. Scheduling Environments
```

Select option **3** under **File** to save the definition and press **Enter**.

```

File Utilities Notes Options Help
*****
* 3 1. New          *      Definition Menu
* 2. Open          *
* 3. Save          *
* 4. Save as      * : 'USER1.SOM.WLM'
* 5. Print        *
* 6. Print as GML * . SOM      (Required)
* 7. Cancel       * .
* 8. Exit         *
*****
following options. . . . — 1. Policies
                          2. Workloads
                          3. Resource Groups
                          4. Service Classes
                          5. Classification Groups
                          6. Classification Rules
                          7. Report Classes
                          8. Service Coefficients/Options
                          9. Application Environments
                          10. Scheduling Environments

```

The following screen is displayed.

```

File Utilities Notes Options Help
-----
WLM                               Definition Menu
Command ==> _____

Definition data set . . : 'USER1.SOM.WLM'

Definition name . . . . . SOM      (Required)
Description . . . . . _____

Select one of the
following options. . . . — 1. Policies
                          2. Workloads
                          3. Resource Groups
                          4. Service Classes
                          5. Classification Groups
                          6. Classification Rules
                          7. Report Classes
                          8. Service Coefficients/Options
*****
* Service definition was saved. (IWMAM081) *
*****

```

Installing a Definition

You need to install the service definition onto the WLM couple data set. Installing the service definition makes any changes available for policy activation.

The following tasks will walk you through installing a definition.

Select option **1** under **Utilities** to install the definition and press **Enter**.

```

File Utilities Notes Options Help
----- *****
WLM * 1 1. Install definition * LEVEL003
Comma * 2. Extract definition *
* 3. Activate service policy *
Defin * 4. Allocate couple data set *
* 5. Allocate couple data set using CDS values *
Defin *****
Description . . . . .
Select one of the
following options. . . . — 1. Policies
                             2. Workloads
                             3. Resource Groups
                             4. Service Classes
                             5. Classification Groups
                             6. Classification Rules
                             7. Report Classes
                             8. Service Coefficients/Options
                             9. Application Environments
                             10. Scheduling Environments

```

The following screen is displayed.

```

File Utilities Notes Options Help
----- *****
WLM                               Definition Menu           LEVEL003
Command ==> _____
Definition data set . . : 'USER1.SOM.WLM'
Definition name . . . . . SOM           (Required)
Description . . . . . _____
Select one of the
following options. . . . — 1. Policies
                             2. Workloads
                             3. Resource Groups
                             4. Service Classes
                             5. Classification Groups
                             6. Classification Rules
                             7. Report Classes
                             8. Service Coefficients/Options
                             9. Application Environments
                             10. Scheduling Environments
*****
* Service definition was installed. (IWMAM038) *
*****

```

Activating the Service Policy

You now need to activate the policy. When you select this option, the application displays a list of the service policies defined in the service definition currently installed on the WLM couple data set. You activate the service policy by selecting it from the list.

The following tasks will walk you through activating a service policy.

Select option **3** under **Utilities** to activate the service policy and press **Enter**.

Note: This is the same as the operator command VARY WLM,POLICY=SOM30.

```
File Utilities Notes Options Help
----- *****
WLM * 3 1. Install definition * LEVEL003
Comma * 2. Extract definition *
* 3. Activate service policy *
Defin * 4. Allocate couple data set *
* 5. Allocate couple data set using CDS values *
Defin *****
Description . . . . .
Select one of the
following options. . . . . 1. Policies
                           2. Workloads
                           3. Resource Groups
                           4. Service Classes
                           5. Classification Groups
                           6. Classification Rules
                           7. Report Classes
                           8. Service Coefficients/Options
                           9. Application Environments
                          10. Scheduling Environments
```

The following screen is displayed.

Select the desired policy from the list (with a /) and press **Enter**.

```
File Utilities Notes Options Help
- *****
* Policy Selection List Row 1 to 1 of 1 *
C * Command ==> _____ *
* *
D * The following is the current Service Definition installed on the WLM *
* couple data set. *
D * *
D * Name . . . . : SOM *
* *
S * Installed by : USER1 from system SY1 *
f * Installed on : 1996/10/17 at 11:39:51 *
* *
* Select the policy to be activated with "/" *
* *
* Sel Name Description *
* / SOM30 *
* ***** Bottom of data ***** *
* *
* F1=Help F2=Split F5=KeysHelp F7=Up F8=Down *
* F9=Swap F12=Cancel *
*****
```

The following screen is displayed.

```

File Utilities Notes Options Help
-----
WLM                               Definition Menu                LEVEL003
Command ==> _____

Definition data set . . . : 'USER1.SOM.WLM'

Definition name . . . . . SOM          (Required)
Description . . . . . _____

Select one of the
following options. . . . _ 1. Policies
                           2. Workloads
                           3. Resource Groups
                           4. Service Classes
                           5. Classification Groups
                           6. Classification Rules
                           7. Report Classes
                           8. Service Coefficients/Options
                           9. Application Environments
                           10. Scheduling Environments
*****
* Service policy SOM30 was activated. (IWMAM060) *
*****

```

Issue the operator command F WLM,MODE=GOAL to change from compatibility to goal mode. Servers will not start unless the system is in goal mode.

Managing Servers Using WLM Commands

WLM provides commands for displaying and controlling a server. The following section lists those commands and gives examples.

Displaying a Server's Status

The following operator command will cause WLM to display status of a server with an application environment name of NAMINGSERVER.

```
DISPLAY WLM,APPLENV=NAMINGSERVER
```

If the state of the server is not available, the server can not be started until it is made available with a VARY command.

```
SY1 IWM029I 06.54.02 WLM DISPLAY 308
APPLICATION ENVIRONMENT NAME      STATE      STATE DATA
NAMINGSERVER                       AVAILABLE
ATTRIBUTES: PROC=GOSOSSVR         SUBSYSTEM TYPE: SOM
```

Stopping a Server

The following operator command causes WLM to stop accepting new binds for the server with an application environment name of NAMINGSERVER and request that the existing server be stopped.

```
VARY WLM,APPLENV=NAMINGSERVER,QUIESCE
```

Making a Server Available

The following operator command causes WLM to accept new binding requests for the server with an application environment name of NAMINGSERVER.

```
VARY WLM,APPLENV=NAMINGSERVER,RESUME
```

Stopping and Making a Server Available

The following operator command causes WLM to send a request to stop the server with an application environment name of NAMINGSERVER. After the server has been stopped, a new binding request will cause the server to be started.

```
VARY WLM,APPLENV=NAMINGSERVER,REFRESH
```

Chapter 3. Registering Classes and Servers in the Implementation Repository

This chapter describes the implementation repository and how to work with it. Specifically, it describes how to use the REGIMPL utility to manage servers and class associations in the implementation repository.

The implementation repository contains information that describes the locations of servers, the names of classes that each server implements, and the actual names of those servers. Before a server can be used it must be registered in the implementation repository.

The implementation repository is always made up of the following data sets:

- *somdir*.IMPL.DB
- *somdir*.ALIAS.DB
- *somdir*.ALIASEDAT.DB

where *somdir* is the value of the SOMDDIR environment variable. For information on the SOMDDIR environment variable, see “[somd] Stanza (Distributed SOMobjects)” on page 80. The VSAM clusters that contain the implementation repository are allocated on the volume specified by the VOLUME environment variable of the [sombt] stanza. For information on the VOLUME environment variable, see “[sombt] Stanza (BTree Framework)” on page 77.

Using the REGIMPL Utility to Register Servers

The REGIMPL utility registers a server's implementation in the implementation repository. SOMobjects uses the information to initialize the server.

Considerations:

- The SOM subsystem and the naming server must be running prior to registering servers and classes.
- Servers must be registered on the machine on which they will run.
- Client processes no longer consult the implementation repository so if you are replicating the implementation repository instead of using a shared volume, you do not have to replicate the implementation repository on systems where clients are run.

This section describes ways to run REGIMPL:

- If you issue the REGIMPL command without options, the REGIMPL utility prompts you for all the information needed to proceed.

For a complete description, see “Running REGIMPL in Prompting Mode” on page 54

- You can also provide all the information needed by issuing the REGIMPL command with options specified.

For a complete description, see “Running REGIMPL in Fast Path Mode” on page 62

Note: You can also run REGIMPL as a batch job. See SOMMVS.SGOSJCL(GOSREGMP) for more information.

Running REGIMPL in Prompting Mode

Follow this step-by-step procedure when running the REGIMPL utility:

1. Specify, on the SOMDDIR environment variable, the directory in which the implementation repository resides, as described in the section on preparing the distributed SOMobjects environment in *OS/390 V2R4.0 SOMobjects Programmer's Guide*.
2. Ensure that your installation has set up the OpenEdition shell to run REGIMPL interactively. For instructions on how to do this, see Chapter 7, "Preparing the OpenEdition Environment to Run SOMobjects Utilities" on page 95.
3. Enter the following command on the OpenEdition command line:

```
> regimpl
```

The system displays the following panel:

```
GOSD0132I
DSOM IMPLEMENTATION REGISTRATION UTILITY
(C) Copyright IBM Corp. 1992,1997. All rights reserved.

GOSD0130I
[ SERVER OPERATIONS ]
 1. Add      2. Delete  3. Change
 4. Show one 5. Show all 6. List aliases

[ CLASS OPERATIONS ]
 7. Add      8. Delete  9. Delete from all 10. List classes
11. Add to all

12. SAVE and EXIT

Enter an operation: 1
```

Figure 1. REGIMPL Utility Main Panel

The following sections explain how to use the REGIMPL utility to update the contents of your implementation repository.

Adding a Server to the Implementation Repository

From the REGIMPL main panel, do the following to add a server to the implementation repository:

1. Enter **1** at the *Enter an operation* prompt (shown in Figure 1).
2. Answer the prompts from REGIMPL as follows:

Implementation alias

Enter a "shorthand" name for conveniently referencing the server, as shown in the following example:

```
GOSD0078I Enter an alias for a new server :CARSERVER1
```

The alias name must conform to the rules for specifying a WLM application environment name. These rules are:

- a. It cannot be NULL or all blanks.
- b. It cannot have embedded blanks in it (for example: moo moo).
- c. Valid characters are alphanumeric, national (@,#,\$), and the underscore characters.
- d. It cannot exceed 32 characters.
- e. It cannot start with 'SYS'.

The alias name specified must be unique, since the alias name is stored in upper case characters in the implementation repository. For example, if the user were to type in carserver and then type in CarServer, the second carserver would fail because it already is stored in the implementation repository as CARSERVER.

ImplementationDef class

Enter the name of the **ImplementationDef** class for REGIMPL to use for this entry (the default is **ImplementationDef**). If a user-defined subclass of **ImplementationDef** was chosen for this server, REGIMPL will then prompt for additional information as defined by the **ImplementationDef** subclass.

Note: If you do not enter the name of a class or subclass, REGIMPL uses the default **ImplementationDef** class.

```
GOSD0397I Enter ImplDef class name (default: ImplementationDef):
```

Server authentication

Specify whether the server should accept requests from authenticated clients only.

Note: If you do not enter **y** or **n**, REGIMPL uses the default value of **n**, which means that the server is not secure.

```
GOSD0460I Make server secure [y/n] (default: no):
```

Server class

By default, REGIMPL uses the SOMDServer class. If your program uses object services, specify the object services server class (somOS::Server). For more information, see *OS/390 V2R4.0 SOMobjects Object Services*.

The following example illustrates how to specify the server class:

```
GOSD0083I Enter server class name (default: SOMDServer) :
```

Select protocol

Select the communications protocol that this server will support.

Note: If you do not enter **y** or **n**, REGIMPL uses the default value of **y**, which means that the server will support the listed protocol.

The following example illustrates how to select the protocol:

```
GOSD0458I Select protocol 'SOMD_TCPIP'? [y/n] (default: yes):
```

3. REGIMPL next displays a summary of the information you entered and asks you to confirm that the information is valid:

```
GOSD0537I
=====
Implementation ID.....: 325ed4d8-0003fe34-7f-00-095f015237e9
Implementation alias.....: CARSERVER1
ImplDef class name.....: ImplementationDef
Server secure.....: No
Server class.....: SOMDServer
Protocol information.....:
    Protocol: SOMD_TCPIP;Hostname: mvs.sys.com;Port: 9393;
=====
GOSD0125I The above server is about to be added. Add? [y/n]  y
```

4. Enter **y** to save the server implementation in the implementation repository or **n** to discard the information and tell REGIMPL to display the main panel again. If you decide to save the implementation, REGIMPL displays the following message:

```
GOSD0111I Server 'CARSERVER1' was successfully added.
```

REGIMPL then displays the main panel shown in Figure 1 on page 54.

Changing a Server in the Implementation Repository

From the REGIMPL main panel, do the following to change a server's definition in the implementation repository:

1. Enter **3** at the *Enter an operation* prompt (shown in Figure 1 on page 54).
2. Enter the alias of the server to update. The following example illustrates the values that can be modified on an update. If nothing is entered at the input prompts, the value that the server initially had before the update will remain in effect.

```
GOSD0077I Enter the alias of the server to update :CARSERVER1
GOSD0118I New alias :CARSERVER2
GOSD0399I New ImplDef class name:
GOSD0463I Make server secure? [y/n]:
GOSD0120I New server class :
GOSD0465I Update protocol information? [y/n] (default: no):
```

3. REGIMPL next displays a summary of the information you entered and asks you to confirm that the information is valid:

```
GOSD0537I
=====
Implementation ID.....: 325ed4d8-0003fe34-7f-00-095f015237e9
Implementation alias.....: CARSERVER2
ImplDef class name.....: ImplementationDef
Server secure.....: No
Server class.....: SOMDServer
Protocol information.....:
    Protocol: SOMD_TCPIP;Hostname: mvs.sys.com;Port: 9393;
=====
GOSD0126I The above server is about to be updated. Update? [y/n]
```

4. Enter **y** to update the server implementation in the implementation repository or **n** to discard the information and tell REGIMPL to display the main panel again. If you decide to update the implementation, REGIMPL displays the following message:

```
GOSD0113I Server 'CARSERVER1' was successfully updated.
```

Show One Server in the Implementation Repository

From the REGIMPL main panel, do the following to show one server on the implementation repository:

1. Enter **4** at the *Enter an operation* prompt (shown in Figure 1 on page 54).
2. Enter the alias of the server to show: The following example illustrates how to show one server:

```
GOSD0076I Enter the alias of the server to show :CARSERVER2
```

3. After pressing Enter, REGIMPL displays the following information:

```

GOSD0537I
=====

Implementation ID.....: 325ed4d8-0003fe34-7f-00-095f015237e9
Implementation alias.....: CARSERVER2
ImplDef class name.....: ImplementationDef
Server secure.....: No
Server class.....: SOMDServer
Protocol information.....:
    Protocol: SOMD_TCPIP;Hostname: mvs.sys.com;Port: 9393;

=====

```

Show All Servers in the Implementation Repository

From the REGIMPL main panel, do the following to show all servers on the implementation repository:

1. Enter **5** at the *Enter an operation* prompt (shown in Figure 1 on page 54).
2. The following results from pressing Enter:

```

GOSD0537I
=====

Implementation ID.....: 32caeeea-0008ad6a-7f-00-328420d03942
Implementation alias.....: MYNAMINGSERVER
ImplDef class name.....: ImplementationDef
Server secure.....: No
Server class.....: somOS::Server
Protocol information.....:
    Protocol: SOMD_TCPIP;Hostname: mvs.sys.com;Port: 9393;

=====

GOSD0537I
=====

Implementation ID.....: 325ed4d8-0003fe34-7f-00-095f015237e9
Implementation alias.....: CARSERVER2
ImplDef class name.....: ImplementationDef
Server secure.....: No
Server class.....: SOMDServer
Protocol information.....:
    Protocol: SOMD_TCPIP;Hostname: mvs.sys.com;Port: 9393;

=====

```

List Aliases

From the REGIMPL main panel, do the following to list the aliases.

1. Enter **6** at the *Enter an operation* prompt (shown in Figure 1 on page 54).
2. The following results from pressing Enter:

```

GOSD0396I
Implementation Alias
=====
MYNAMINGSERVER
CARSERVER2

```


Adding a Class to a Server Implementation

Use the following step-by-step procedure to add a class to a server implementation:

1. From the REGIMPL main panel shown in Figure 1 on page 54, enter **7** at the *Enter operation* prompt.
2. Answer the prompts from REGIMPL as follows:

Class name

Enter the name of a class associated with the implementation alias, as shown in the following example:

```
GOSD0081I Enter the name of a class :Dvehicle
```

Implementation alias

Enter the alias of the server that implements the new class. The following example specifies the alias defined in “Adding a Server to the Implementation Repository” on page 54:

```
GOSD0074I Enter the alias of the server that implements the class :CARSERVER2
```

Additional properties

REGIMPL also lets you associate your own user-defined properties with the name bindings it creates when a class/server pair is registered. These additional properties can then be used in queries to the DSOM Factory Service by client programs. Additional properties are specified to REGIMPL as the property's name, followed by a space, followed by the property value.

Signify no more additional properties by pressing Enter.

```
GOSD0467I Enter additional property <name value> pairs:
```

3. REGIMPL next displays a summary of the information you entered and asks you to confirm that the information is valid:

```
GOSD0538I
=====
Implementation ID.....: 325ed4d8-0003fe34-7f-00-095f015237e9
Implementation alias.....: CARSERVER2
Class name.....: Dvehicle
=====
GOSD0466I The above class is about to be added. Add? [y/n]:
```

4. Enter **y** to associate the class to the server implementation in the naming service or enter **n** to discard the information and display the main panel again. All class/server associations are registered in the Naming Service, for use by the DSOM Factory Service. When a REGIMPL user associates a class with a server, as above, REGIMPL creates a name binding in the Naming Service to reflect that association. If processing is successful, REGIMPL displays the following message:

```
GOSD0069I Class 'Dvehicle' is now associated with server 'CARSERVER2'
```

REGIMPL then displays the main panel shown in Figure 1 on page 54.

Adding a Class to all Servers

Use the following step-by-step procedure to add a class to all servers:

1. From the REGIMPL main panel shown in Figure 1 on page 54, enter **11** at the *Enter operation* prompt.
2. After pressing Enter, REGIMPL responds with the following message:

```
GOSD0081I Enter the name of a class :Car
```

3. After pressing Enter, REGIMPL responds with the following message:

```
GOSD0434I The class 'Car' is now associated with all servers.
```

REGIMPL then displays the main panel shown in Figure 1 on page 54.

Listing Classes in the Implementation Repository

Use the following step-by-step procedure to list classes in the implementation repository.

1. From the REGIMPL main panel shown in Figure 1 on page 54, enter **10** at the *Enter operation* prompt.
2. After pressing Enter, REGIMPL responds with the following message:

```
GOSD0010I
Implementation alias      Associated classes
-----
MYNAMINGSERVER          Car
CARSERVER2              Dvehicle
                       Car
```

REGIMPL then displays the main panel shown in Figure 1 on page 54.

Deleting a Class from All Server Implementations

Use the following step-by-step procedure to delete a class from all server implementations:

1. From the REGIMPL main panel shown in Figure 1 on page 54, enter **9** at the *Enter operation* prompt.
2. Answer the prompts from REGIMPL as follows:

Class name

Enter the name of the class to delete from the Naming Service:

```
GOSD0080I Enter the name of the class to delete :Car
```

3. After pressing Enter, REGIMPL responds with the following message:

```
GOSD0438I Class 'Car' has been removed from the Naming Service.
```

REGIMPL then displays the main panel shown in Figure 1 on page 54.

Deleting One Class from a Server Implementation

Use the following step-by-step procedure to delete a class from a server implementation:

1. From the REGIMPL main panel shown in Figure 1 on page 54, enter **8** at the *Enter operation* prompt.
2. Answer the prompts from REGIMPL as follows:

Class name

Enter the name of the class to delete as shown in the following example:

```
GOSD0081I Enter the name of a class :Dvehicle
```

Implementation alias

Enter the alias of the server that implements the class that is being deleted. The following example specifies the alias defined in “Adding a Server to the Implementation Repository” on page 54:

```
GOSD0074I Enter the alias of the server that implements the class :CARSERVER2
```

REGIMPL responds with the following message:

```
GOSD0436I The class 'Dvehicle' has been removed from server CARSERVER2.
```

REGIMPL then displays the main panel shown in Figure 1 on page 54.

Deleting a Server in the Implementation Repository

From the REGIMPL main panel, do the following to delete a server on the implementation repository:

1. Enter **2** at the *Enter an operation* prompt (shown in Figure 1 on page 54).
2. Answer the prompts from REGIMPL as follows:

Delete server

Enter the alias of the server to delete:

The following example illustrates how delete the alias of the server:

```
GOSD0075I Enter the alias of the server to delete :CARSERVER2
```

After following the prompts, you will receive the next message:

```
GOSD0112I Server 'CARSERVER2' was successfully deleted.
```

Saving Your Data and Exiting REGIMPL

Use the following step-by-step procedure to save your data and exit REGIMPL:

1. From the REGIMPL main panel shown in Figure 1 on page 54, enter **12** at the *Enter operation* prompt.
2. The REGIMPL utility ends with the following message:

```
GOSD0440I REGIMPL processing has completed.
```

| Running REGIMPL in Fast Path Mode

REGIMPL has a command line interface that offers the same functions as the REGIMPL utility, which is described in “Using the REGIMPL Utility to Register Servers” on page 53. You can enter REGIMPL commands at the system prompt or in batch files.

REGIMPL Help Screen

The following is the help screen for REGIMPL command. To get to the help screen, enter:

```
> regimpl -?
```

```

GOSD0132I
DSOM IMPLEMENTATION REGISTRATION UTILITY
(C) Copyright IBM Corp. 1992,1997. All rights reserved.

GOSD0020I
To enter interactive mode:
  regimpl
Add implementation:
  regimpl -A -i <str> [-v <str>]
           [-z <str>]
           [-e <str> {<param> ...}] [-s {on|off}]
           [-t <str> {<param> ...} [-t ...]]
Update implementation:
  regimpl -U -i <str> [-v <str>]
           [-e <str> {<param> ...}] [-s {on|off}]
           [-t <str> {<param> ...} [-t ...]]
Delete implementation:
  regimpl -D -i <str> [-i ...]
List implementation(s):
  regimpl -L [-i <str> [-i ...]]
List aliases:
  regimpl -S
Add class(es):
  regimpl -a -c <str> [-c ...] -i <str> [-i ...]
Delete class(es):
  regimpl -d -c <str> [-c ...] [r> []]
List classes associated with implementation(s):
  regimpl -l [-i <str> [-i ...]]
where:
-c <str>    = Class name
-e <str>    = ImplDef Class name (default: ImplementationDef)
  <param>   = Values for additional attributes needed by subclass of
             ImplementationDef
-i <str>    = Implementation alias name
-s {on|off} = Enable secure server (optional)
-t <str>    = Transport protocol
  <param>   = Option can be zero or more strings, delimited by
             spaces.
-v <str>    = Server-class name (default: SOMDServer)
-z <str>    = Implementation ID

```

Figure 2. REGIMPL Help Screen

REGIMPL Command Syntax

The following are the forms of the REGIMPL command. The parameters are described after the syntax.

regimpl

Tells REGIMPL to enter interactive mode, which is described in “Running REGIMPL in Prompting Mode” on page 54.

regimpl -A

Adds an implementation. The syntax of the command is:

```

regimpl -A -i <str> [-v <str>]
                [-z <str>]
                [-e <str> {<param> ...}] [-s {on|off}]
                [-t <str> {<param> ...} [-t ...]]

```

regimpl -U

Updates an implementation. The syntax of the command is:

```
regimpl -U -i <str> [-v <str>]
                    [-e <str> {<param> ...}] [-s {on|off}]
                    [-t <str> {<param> ...} [-t ...]]
```

regimpl -D

Deletes one or more implementations. The syntax of the command is:

```
regimpl -D -i <str> [-i ...]
```

regimpl -L

Lists one or more implementations. The syntax of the command is:

```
regimpl -L [-i <str> [-i ...]]
```

regimpl -S

Lists all implementation aliases. The syntax of the command is:

```
regimpl -S
```

regimpl -a

Adds class associations to one or more implementations. The syntax of the command is:

```
regimpl -a -c <str> [-c ...] -i <str> [-i ...]
```

regimpl -d

Deletes class associations from one or more implementations. The syntax of the command is:

```
regimpl -d -c <str> [-c ...] [-i <str> [-i ...]]
```

regimpl -l

Lists classes associated with one or more implementations. The syntax of the command is:

```
regimpl -l [-i <str> [-i ...]]
```

REGIMPL Example 1: To Add a Server

To add a server, such as CARSERVER1, enter the following:

```
regimpl -A -i CARSERVER1
```

REGIMPL Example 2: To Add a Class to a Server

To add a class (such as Dvehicle) to a server (such as CARSERVER1), enter the following:

```
regimpl -a -i CARSERVER1 -c Dvehicle
```

Chapter 4. Configuring and Administering Security for SOMobjects

Because many different systems and users can be involved in distributed communications, SOMobjects provides several ways to protect against unauthorized access.

SOMobjects provides services for authorizing and authenticating clients in a *trusted network*. IBM does not recommend that you use SOMobjects in a network that is not trusted, because SOMobjects does not provide encryption and verification of transmitted data. Note that if RACF PassTickets are not used, the transmitted data can include user IDs and their associated passwords.

When setting up SOMobjects, all installations must take the following steps:

- ___ 1. Associate user identities for the SOM subsystem and SOMobjects servers
- ___ 2. Authorize SOMobjects servers to the SOM subsystem
- ___ 3. Ensure that SOMobjects servers and clients have OpenEdition access
- ___ 4. If the RACF APPL class is active, ensure that SOMobjects clients are authorized to SOM in the APPL class.

When you set up SOMobjects servers, you can configure them to run in “secure mode.” If you do so, you must also:

- ___ 1. Ensure that any clients intending to use the secure server are authenticated clients. (This can be done by specifying valid USER and PASSWD in their configuration files and by creating RACF user profiles for them in the install host. For OS/390 clients, you can also set up RACF PassTickets, which are a more secure substitute for passwords.)
- ___ 2. Ensure that these clients are authorized to use the secure SOMobjects server.

Security Considerations for Non-Secure Servers

For any non-secure SOMobjects server (any server that is **not** running in secure mode), note the following:

- Clients that run unauthenticated can use the server. This means that you might not be able to identify the persons who actually invoke applications that use the server.
- Clients can run unauthenticated by specifying `DISABLE_AUTHN=TRUE` in their local configuration file.

If you use the `SOMM_MVS_Secure` metaclass, you must also authorize appropriate clients to use the protected classes and methods.

Hints & Tips

Use RACF Groups to Simplify RACF Administration

To simplify RACF administration for SOMobjects, IBM recommends that you set up one or more RACF groups to meet your specific requirements. Some examples:

- A RACF group for each secure server
- A RACF group for all OS/390 clients
- A RACF group for all non-OS/390 clients
- A RACF group for SOMobjects administrators
- A RACF group for each related set of secure classes and methods.

If you use RACF groups, you can give users access to RACF-protected resources by connecting them to the groups (CONNECT command) rather than by adding individual user IDs to access lists and (for some RACF classes), issuing the SETROPTS RACLIST REFRESH command for the class.

Terminology: “Principal”

A *principal* represents the identity of a user (or other entity) invoking a method. In SOMobjects, “principal” and “user ID” are equivalent terms.

Before You Begin

Before you begin to set up security for SOMobjects, determine if the Security Server (RACF) or equivalent is installed on your OS/390 system.

If RACF is *not* installed, SOMobjects can still be used. However, there are minimal access controls.

If RACF *is* installed, verify that the level of RACF supports the CBIND, SERVER, SOMDOBJ, and GSOMDOBJ general resource classes. One way to do this is to issue the SETROPTS LIST command.

Auditing Considerations

You might be interested in auditing who uses SOMobjects classes, methods, and servers. One way to do this is to set up RACF auditing. RACF provides auditing support for the APPL, CBIND, SERVERS, SOMDOBJ, and GSOMDOBJ classes and profiles, and for OS/390 clients. For more information on RACF auditing, see *OS/390 Security Server (RACF) Auditor's Guide*.

Security Configuration for All SOMobjects Installations

The following sections describe how to set up security for all SOMobjects installations:

- “Associating User IDs with the SOM Subsystem and SOMobjects Servers” on page 69

- “Authorizing SOMobjects Servers to the SOM Subsystem” on page 70
- “Authorizing SOMobjects Servers and Clients to OpenEdition” on page 70
- “Authorizing Clients to SOMobjects” on page 72

Associating User IDs with the SOM Subsystem and SOMobjects Servers

When setting up SOMobjects, you must associate user IDs with the SOM subsystem and SOMobjects servers.

1. Enter the following RACF commands:

```

SETROPTS CLASSACT(STARTED)
SETROPTS RACLIST(STARTED)
SETROPTS GENERIC(STARTED)
RDEFINE STARTED SOM.* STDATA(USER(SOM) GROUP(SYS1))
RDEFINE STARTED GOSOSSVR.* STDATA(USER(SOM) GROUP(SYS1))
RDEFINE STARTED GOSSVR.* STDATA(USER(SOM) GROUP(SYS1))
SETROPTS RACLIST(STARTED) REFRESH

```

where

SOM is the name of the cataloged procedure used for the SOM subsystem

GOSOSSVR is the name of the cataloged procedure used for the naming server and security server, and any other object services servers.

GOSSVR is the name of the cataloged procedure used for application servers.

Note: If specific servers have other security requirements, consider creating unique cataloged procedures for them, and associating unique user identities with those servers.

2. For backup purposes, in case the STARTED class is inadvertently deactivated, IBM recommends also updating the RACF started procedures table (ICHRIN03 module). To do this, add the following assembler statements to your existing RACF started procedures table (ICHRIN03 module):

```

DC CL8'SOM' Procedure name
DC CL8'SOM' Userid
DC CL8'SYS1' Group name
DC XL1'00' Attribute bits
DC XL7'00' Reserved
DC CL8'GOSOSSVR' Procedure name
DC CL8'SOM' Userid
DC CL8'SYS1' Group name
DC XL1'00' Attribute bits
DC XL7'00' Reserved
DC CL8'GOSSVR' Procedure name
DC CL8'SOM' Userid
DC CL8'SYS1' Group name
DC XL1'00' Attribute bits
DC XL7'00' Reserved

```

Then assemble and link ICHRIN03 into the system.

3. Give these user IDs WRITE access to all data sets with a high-level qualifier of SOMMVS.

4. Give the SOM user ID access to OpenEdition. For specific instructions, see *OS/390 OpenEdition Planning*.
5. Add additional RACF definitions for each additional SOMObjects server, as needed.

For more information, see *OS/390 Security Server (RACF) Security Administrator's Guide*.

Authorizing SOMObjects Servers to the SOM Subsystem

When a SOMObjects server is started, SOMObjects verifies the server's authorization to communicate with the SOM subsystem. To control access, issue the following RACF commands:

```
SETROPTS CLASSACT(SERVER)
SETROPTS RACLIST(SERVER)
SETROPTS GENERIC(SERVER)
RDEFINE SERVER SOM.SOM.CARSERVER1
PERMIT SOM.SOM.CARSERVER1 CLASS(SERVER) +
      ID(server-userid) ACCESS(READ)
SETROPTS RACLIST(SERVER) REFRESH
```

where *SOM* is the subsystem name, and *CARSERVER1* is the server_alias_name.

The maximum size for *server_alias_name* is 32 characters.

The same authorization checks are performed when the server “unregisters” with SOMObjects, or when a server cannot complete initialization.

Authorizing SOMObjects Servers and Clients to OpenEdition

To use SOMObjects, the following user IDs must have access to OpenEdition services:

- The user ID associated with the SOM subsystem
- The user ID(s) associated with SOMObjects servers
- The user IDs associated with SOMObjects clients
- The user IDs of the users that run the REGIMPL utility.

Configure the user IDs that are associated with the SOM subsystem and the SOMObjects server address spaces for OpenEdition. This means that they must have UID, HOME, and PROGRAM information specified, and they must be associated with a group that has a GroupID (GID) specified.

For example, to define USER1 to OpenEdition, issue the following RACF command:

```
ALTUSER USER1 OMVS(UID(uid) +
      HOME(home_path_name) +
      PROGRAM(shell_program_name))
```

For information about how to provide OpenEdition access, see the section on defining OpenEdition users in *OS/390 OpenEdition Planning*.

Running a Server in Secure Mode

A server running in secure mode accepts method requests from authenticated clients only. Authenticated clients are:

- Clients that have supplied a valid user ID and password (as specified in the USER and PASSWD settings in the user's configuration file). For more information, see Chapter 5, “Setting up Configuration Files” on page 75.
- Clients that have supplied a valid RACF PassTicket. For OS/390 clients, this is the preferred method of authentication.

Configuring a Server to Run in Secure Mode

To configure a server to run in secure mode, change their SOMobjects registration using the REGIMPL utility. Choose option 3 (Change) and answer *y* to the prompt Make server secure?. For more information on changing a server's registration, see “Changing a Server in the Implementation Repository” on page 56.

If the server is running in secure mode, only authenticated clients can invoke methods. If, for any reason, the setting of the secure flag is changed, the application server must be restarted for the new setting to take effect. For more information, see “Stopping and Making a Server Available” on page 51.

Authorizing Clients to Secure Servers

After a client is authenticated, if the server is running in secure mode (*-s* on specified on the REGIMPL command), SOMobjects checks the authorization of the client to bind with the requested server.

To control this access, give clients READ access authority to a profile named SOM.subsystem-name.server-alias-name in the CBIND RACF general resource class. For example, if the subsystem-name is SOM and the server alias name is CARSERVER1, issue the following RACF commands:

```
RDEFINE  CBIND SOM.SOM.CARSERVER1
PERMIT  SOM.SOM.CARSERVER1 +
        CLASS(CBIND) ID(userid) ACCESS(READ)
SETROPTS RACLIST(CBIND) REFRESH
```

When a client connects to a secure server, the server ensures that the client has passed through the necessary checks by the SOM subsystem. If the test is successful, the client obtains access to the server. If the server uses method level authorization functions (described in the metaclass chapter of *OS/390 V2R4.0 SOMobjects Programmer's Guide*), the server verifies that the client can access requested methods, using the SOMDOBJs RACF class.

Authorizing Clients to SOMobjects

During client authentication, SOMobjects asks RACF to verify that the user ID and password supplied by the client are correct. In addition, SOMobjects verifies that the client has authorization to use SOMobjects by checking the user's authorization to the application SOM in the class APPL.

This check is performed only when the APPL class is active.

To control access to SOMobjects and allow specific users or groups access, issue the following RACF commands:

```
SETROPTS CLASSACT(APPL)
SETROPTS RACLIST(APPL)
RDEFINE APPL SOM
PERMIT SOM CLASS(APPL) +
      ID(userid or group-name) ACCESS(READ)
SETROPTS RACLIST(APPL) REFRESH
```

Setting up the PassTicket Environment

If the client and the server are both running on OS/390, you can use RACF PassTickets as an alternative to specifying the PASSWD environment variable. RACF PassTickets are cryptographically-generated, single-use, short-lifespan password substitutes. They are inherently more secure than passwords.

IBM recommends that you use RACF PassTickets to authenticate clients.

To define the PassTicket environment, you need to define a shared key or secret between the systems on which the client and the security server are running. This is done by defining the PassTicket environment on both systems. If both the client and the security server are running on the same system, then the commands must be entered on only that system.

To create the RACF PassTicket environment for SOM with the shared key of X'1234567890ABCDEF', take the following steps:

1. Activate PassTickets on the host system, and on any OS/390 system on which distributed clients will be running:

```
SETROPTS CLASSACT(PTKTDATA)
SETROPTS RACLIST(PTKTDATA)
RDEFINE PTKTDATA SOM SSIGNON(KEYMASKED(1234567890ABCDEF))
SETROPTS RACLIST(PTKTDATA) REFRESH
```

2. Ensure that the PASSWD environment variable is removed from any configuration file used by any client that uses PassTickets.
3. To prevent PassTicket collisions, ensure that the SOM subsystem, each SOM server, and each client that uses PassTickets has a unique security identity.

This include batch jobs that are SOMobjects clients. For example, if a user submits two batch jobs that run as SOMobjects clients, and those batch jobs do **not** have the USER parameter specified on the JOB statement, then those two

batch jobs have the same security identity, and a PassTicket collision could occur. To prevent this from happening, do one of the following:

- Make sure the batch jobs run one at a time.
- Submit the jobs under unique user identities (for example, using surrogate user support; for more information, see *OS/390 Security Server (RACF) Security Administrator's Guide*).

Authorizing Clients to Specific Methods

To enable the method-level access control, define profiles in the RACF classes SOMDOBJ and GSOMDOBJ.

Profiles in the SOMDOBJ class must be defined as *classname.methodname*, where *classname* is the SOM class of the object receiving the request to invoke the method identified by *methodname*.

Use profiles in the GSOMDOBJ class for classes and methods that have dissimilar names. Profile names in the GSOMDOBJ class can be any name allowed by the RACF RDEFINE command, but the members specified on the ADDMEM operand must observe the naming conventions for profiles in the SOMDOBJ class *classname.methodname* as in the SOMDOBJ profile names.

Notes:

1. IBM recommends that you avoid defining GSOMDOBJ profiles such that they protect classes and methods that are also protected by SOMDOBJ profiles.
2. The maximum size SOMobjects allows is 123 bytes for the class name and 123 bytes for the method name.

For example, to protect access to the method START that is applicable to objects of class DVEHICLE, create a RACF profile named DVEHICLE.START in the SOMDOBJ general resource class, then give the appropriate users or groups READ access authority:

```
SETROPTS CLASSACT(SOMDOBJ)
SETROPTS RACLIST(SOMDOBJ)
SETROPTS GENERIC(SOMDOBJ)
RDEFINE SOMDOBJ DVEHICLE.START
PERMIT DVEHICLE.START CLASS(SOMDOBJ) +
      ID(userid or group-name) ACCESS(READ)
SETROPTS RACLIST(SOMDOBJ) REFRESH
```

For more information about establishing metaclass security, see the description of the SOM_MVS_Secure class in *OS/390 V2R4.0 SOMobjects Programmer's Reference, Volume 1*.

Hints & Tips

RACF always uses the most specific profile that fits a resource. So if the profile MOODYDOG.* is defined and the profile MOODYDOG.HOWL is also defined, and the howl method is invoked on the MOODYDOG class (with method protection enabled) then the MOODYDOG.HOWL profile will be used to check the user's method authorization. For specific information on which profiles are used by RACF when checking authorizations, see the description of generic profile checking of general resources in *OS/390 Security Server (RACF) Security Administrator's Guide*.

If a particular method is protected by both a SOMDOBJ profile and a GSOMDOBJ profile, special rules apply to the access authorizations RACF grants. Look for information on resolving conflicts among multiple profiles in *OS/390 Security Server (RACF) Security Administrator's Guide*.

Creating Profiles in the GSOMDOBJ Class

If you had methods A and B in class X, and methods C and D in class Y. You could create a profile with a name you choose, such as TEST35:

```
RDEFINE GSOMDOBJ TEST35 UACC(NONE)
      ADDMEM(X.A X.B Y.C Y.D)
```

To allow group FINANCE to use these method, enter:

```
PERMIT TEST35 CLASS(GSOMDOBJ) ID(FINANCE) ACCESS(READ)
```

Note: After creating or changing a GSOMDOBJ profile, you must request SETROPTS RACLIST processing for the SOMDOBJ class to make the changes effective on the system.

To protect another method (method M in class Z) with the same profile, change the TEST35 profile as follows:

```
RALTER GSOMDOBJ TEST35 ADDMEM(Z.M)
SETROPTS RACLIST(SOMDOBJ) REFRESH
```

To stop protecting method M in class Z with this profile, change the TEST35 profile as follows:

```
RALTER GSOMDOBJ TEST35 DELMEM(Z.M)
SETROPTS RACLIST(SOMDOBJ) REFRESH
```

Chapter 5. Setting up Configuration Files

This section describes configuration files, gives the syntax for configuration files, and describes the stanzas and keyword variables for SOMobjects.

About Configuration Files

Note: The configuration file is also known as the environment file, the configuration file, and the profile data set.

Configuration File Variable Settings

SOMobjects uses variable settings specified by a configuration file. The configuration file is specified in one of two ways:

- For batch jobs, started tasks, and TSO/E users, the configuration file is an MVS data set specified on the SOMENV DD statement or using the TSO/E ALLOCATE FILE(SOMENV) command. If more than one data set is specified, the first occurrence of any one SOMobjects environment variable is used.

For example, the following illustrates how to specify configuration files through the SOMENV DD statement:

```
//SOMENV DD DSN=SOMMVS.SOMENV.INI,DISP=SHR
```

or

```
//SOMENV DD DSN=MYINI.SOMENV.INI,DISP=SHR  
//      DD DSN=SOMMVS.SOMENV.INI,DISP=SHR
```

- For jobs or users running in an OpenEdition environment, the SOMENV environment variable, if specified, is used to define which file or files are to be used as the configuration file. If the SOMENV environment variable is not specified, the SOMENV DD statement or TSO/E ALLOCATE command is used.

For example, the following illustrates how to specify configuration files through the SOMENV OpenEdition environment variable:

```
SOMENV=//'sommvs.somenv.ini'
```

or

```
SOMENV=//'myini.somenv.ini';//'sommvs.somenv.ini'
```

Note: If both the SOMENV environment variable and the SOMENV DD statement (or TSO/E ALLOCATE command) are specified, the SOMENV environment variable is used and the other is ignored.

The configuration file is organized into stanzas that contain keywords. For example, the SOMDPORT keyword, specified in the [somd] stanza:

```
[somd]  
SOMDPORT=9393
```

The combination of stanza and keyword creates a unique variable setting. The same keyword specified in a different stanza represents a different variable setting. For example:

```
[somd]  
SOMDPORT=9393
```

```
and
[SOMD_TCPIP]
SOMDPÖRT=9393
```

represent two distinct variable settings.

Note: Stanza and keyword names are case sensitive.

When you specify more than one configuration file, the variable settings are combined, starting with the first file or data set. In the case of a duplicate variable specification, the first occurrence of a given variable specification takes precedence.

For example, if data set 'MYINI.SOMENV.INI' contains:

```
[sömd]
SOMDPÖRT=9393
```

and 'SÖMMVS.SÖMENV.INI' contains

```
[sömd]
SOMDPÖRT=5001
```

and your configuration file specification is

SÖMENV=// 'MYINI.SÖMENV.INI' ; // 'SÖMMVS.SÖMENV.INI' , the effective SOMDPÖRT setting for the [sömd] stanza is 9393.

Comparison of Global and Local Configuration File Values

The SOM subsystem processes a configuration file (called the “global configuration file”) during its initialization. The variable settings from that configuration file are available to all SOMobjects applications running on the same system (or in the same sysplex) as the SOM subsystem. The SOM subsystem's configuration file is known as the global configuration file.

Each SOMobjects application can also specify its own configuration file (called the local configuration file). The local configuration file is optional.

When a SOMobjects application specifies a local configuration file, the variable settings in the local configuration file override duplicate variable settings from the global configuration file. For example, if a SOMobjects application's local configuration file contains:

```
[sömc]
SMINCLUDE=//MYAPP.IDL; // 'SÖMMVS.SGÖSIDL' ; // 'SÖMMVS.SGÖSEFW' ;
```

and the global configuration file contains:

```
[sömc]
SMINCLUDE=// 'SÖMMVS.SGÖSIDL' ; // 'SÖMMVS.SGÖSEFW' ;
```

the effective SMINCLUDE setting for that SOMobjects application would be:

```
SMINCLUDE=//MYAPP.IDL; // 'SÖMMVS.SGÖSIDL' ; // 'SÖMMVS.SGÖSEFW' ;
```

Configuration File Syntax

There are two elements present in a configuration file:

- stanza name
- keyword variable name and setting.

These are the syntax rules for the configuration file:

- Comment lines begin with a semicolon (;) in the first position of the line.
- Each stanza begins with the name of the stanza enclosed in square brackets ([and]) at the beginning of a line. The stanza names are case sensitive.
Note: The hexadecimal representation of the left and right square brackets used by SOMobjects is X'AD' and X'BD', respectively.
- Blanks and spaces can appear anywhere in the file.
- Each keyword variable is associated with the stanza name preceding it in the file.
- Keyword variable settings are of the form **Keyword=value**. The keyword variable names are case sensitive. If you wish to continue the value on the subsequent line, put a backward slash (\) as the last non-blank character on the line to be continued and begin the continuation in the first position of the next line.

Stanzas and Keyword Variables

The following sections describe the SOMobjects environment variables. For each variable, the following information is provided:

Usage	Must the variable be explicitly specified in the configuration file? What are the special considerations that would cause it to be required?
Scope	System, local, or both. System means that the variable is defined in the global configuration file and it is not intended that individual SOMobjects processes or users modify such variables. Local means that individual SOMobjects processes or users may find it useful to specify the variable, and the variable isn't intended to affect all processes or users.
Default	The value assigned to the variable if no value is specified in the configuration file.

[sombt] Stanza (BTree Framework)

This stanza specifies the settings for the BTree framework, which provides data management services for SOMobjects.

VOLUME=*valid1 valid2 valid3 valid4 valid5 valid6 valid7*

The VOLUME specification for SOMobjects-related VSAM clusters. If specified, the data from this variable is used as the VOLUMES parameter of the DEFINE CLUSTER command used to create the VSAM data sets. For sysplex considerations, see "Data Sets to Put on Shared Volumes" on page 91.

Usage: Optional if VSAM allocation is SMS-managed. Required if VSAM allocation is not SMS-managed.

Scope: System

Default: If VSAM clusters are SMS-managed, the VOLUME setting is not required and does not affect actual allocation. If SMS is not configured to manage VSAM allocation, the VOLUME setting is required.

One to seven volumes can be specified. If more than one volume is specified, all of them must be of the same device type. For further information on the VOLUMES parameter of the DEFINE CLUSTER command, see *DFSMS/MVS Access Method Services for ICF*.

[somc] Stanza (SOMobjects Compiler)

This stanza specifies the settings for the SOMobjects compiler.

SMOE=YES

Indicates that the SOMobjects compiler should interpret any ambiguous file names as HFS files (files in the hierarchical file system). The default is for ambiguous file names to be interpreted as MVS data set names.

Note: If you are planning to develop SOMobjects applications completely within the OpenEdition shell and using the hierarchical file system, then you should export the SMOE environment variable in your setup or profile shell script. For more information, see *OS/390 V2R4.0 SOMobjects Programmer's Guide*.

Usage: Required if running in the OpenEdition shell; not required otherwise.

Scope: Local

Default: NO

SMINCLUDE=*name1;name2;...namen*

Specifies where the SOMobjects compiler should look for IDL members #included by the IDL data set being compiled.

The values specified for *name1*, *name2*, and so forth can be any of the following:

- MVS data set names (PDSs only, without member names), either fully-qualified or not.
- HFS directories
- A dot (indicates the current directory).

Semicolons separate names. You can specify both HFS directories and MVS data set names, in any order, in the same SMINCLUDE setting. When specifying *name* for the SMINCLUDE variable, the following rules apply:

- To specify a fully-qualified MVS data set name, specify // before the name and put quotes around the name, such as:

```
//'SOMMVS.DSNAME'
```

- To use the current userid (or prefix) as the high-level qualifier, specify // before the name and omit the quotes, such as:

```
//DSNAME
```

- If the IDL member is an HFS file, omit // and specify the path to the directory where the file resides, such as:

```
/u/user1/
```

or, for the current directory, specify a dot (.), such as:

Examples:

To look first in the current directory, then in the IBM-supplied data sets 'SOMMVS.SGOSIDL' and 'SOMMVS.SGOSEFW', specify:

```
SMINCLUDE=.;//'SOMMVS.SGOSIDL';//'SOMMVS.SGOSEFW';
```

To look first in a data set named MYAPP.IDL allocated under the current userid or prefix, then in the IBM-supplied data sets 'SOMMVS.SGOSIDL' and 'SOMMVS.SGOSEFW', specify:

```
SMINCLUDE=//MYAPP.IDL;//'SOMMVS.SGOSIDL';//'SOMMVS.SGOSEFW';
```

Usage: Required only if a configuration file is input to the SOMobjects compiler.

Scope: Both

Default: No default

NEWEMITEFW=*hlq*

Sets the high-level qualifiers found on the profile NEWEMITEFW record into the symbol table referenced by the emitter framework utilities. This value, named "TemplateHLQs" is substituted in the GEN@EMIT template to specify the high-level qualifiers of the user's template.

Usage: Required if running the NEWEMIT utility

Scope: Local

Default: Null

SMNOPRINTNAME=SUPPRESS

Suppresses the printing of processing messages.

Usage: Optional

Scope: Local

Default: No default

SMEMITPREFIX=*emitter-prefix*

Emitter prefix.

Usage: Optional

Scope: Local

Default: emit

SMEMIT=*emitters*

Specifies which output files the SOMobjects compiler produces. Specify a list of items separated by colons. Each item designates an emitter to execute. For example, the statement:

```
SMEMIT=c:h:ih
```

indicates the C, H and IH emitters will be run. This directs the SOMobjects compiler to produce the C implementation template (*dataset_stem.c(payload)*), and the C language bindings (*dataset_stem.h(payload)* and *dataset_stem.ih(payload)*) from the *dataset_stem.idl(payload)* IDL.

By comparison:

```
SMEMIT=xc:xh:xih
```

indicates the XC, XH and XIH emitters will be run. This directs the SOMobjects compiler to produce the C++ implementation template (*dataset_stem.cxx*(payroll), and the C++ language bindings (*dataset_stem.xh*(payroll) and *dataset_stem.xih*(payroll) from the *dataset_stem.idl*(payroll) IDL.

Usage: Optional

Scope: Local

Default: h:ih

SMADDSTAR

When defined, causes all interface references to have a * added to them for the C emitters (c, h, ih). However, the command line SOM compiler options **-maddstar** and **-mnoaddstar** supersede and override the SMADDSTAR setting.

Note: The SMADDSTAR environment variable does not affect output from the C++ emitters (xc, xh, xih, hh). These C++ emitters cause all interface references to have a * added to them.

Usage: Optional

Scope: Local

Default: None (* not added to C)

[somd] Stanza (Distributed SOMobjects)

This stanza specifies the settings for distributed SOMobjects (DSOM).

SOMDDIR=*//hlq*

The high-level qualifier for SOMobjects-related data sets. If you do not want your files to have the high-level qualifier SOMMVS, change the variable, but leave the //. There can be no HFS specification for this variable. Only MVS data set file stems are allowed. SOMDDIR is used for creating and accessing the VSAM files used by SOMobjects.

Usage: Optional

Scope: System

Default: Value of SOMBASE environment variable

SOMDCOMMDEBUGFLAG=*flag*

If SOM_TraceLevel=2 or higher is specified, SOMDCOMMDEBUGFLAG specifies the amount of communications debugging information provided by SOMobjects. Normally, this variable would not be set.

Specifying SOMDCOMMDEBUGFLAG=1 enables a display of TRANSPORT information. TRANSPORT information is data associated with the communication socket such as "Create Listener" and "Sending Message".

Specifying SOMDCOMMDEBUGFLAG=2 enables a display of DATASTREAM information. DATASTREAM information is the actual data sent via a distributed request.

Specifying SOMDCOMMDEBUGFLAG=4 enables a display of DISPATCH information. DISPATCH information is the dispatch of a distributed method request by a SOMobjects server.

To trace multiple types of debugging information, add the flags mathematically, and specify the total. For example, specifying SOMDCOMMDEBUGFLAG=3 enables a display of both TRANSPORT and DATASTREAM information. Specifying SOMDCOMMDEBUGFLAG=7 enables a display of all information above.

Usage: Optional

Scope: Local

Default: If SOM_TraceLevel is 2 or higher, the default is 7. Otherwise, the default is 0.

HOSTNAME=*mvs.sys.com*

The name of the local machine that is running SOMobjects. HOSTNAME can be set only in configuration files used by the SOM subsystem.

Usage: Required

Scope: System

Default: No default.

SOMDPROTOCOLS=*SOMD_TCPIP*

The names of the SOMobjects communications protocols for which the machine has been configured. The only valid value is SOMD_TCPIP.

Usage: Optional

Scope: System

Default: SOMD_TCPIP

SOMDRECVWAIT=*number-of-seconds*

The number of seconds to wait for a socket to become readable before generating a communications timeout error.

Usage: Optional

Scope: Both

Default: 30

SOMDSENDWAIT=*number-of-seconds*

The number of seconds to wait for a socket to become writable before generating a communications timeout error.

Usage: Optional

Scope: Both

Default: 30

MAXIMUMHOPS=*number-of-hops*

The maximum number of location forwarding requests that will be processed before raising an exception.

Usage: Optional

Scope: Local

Default: 1

SOMDTIMEOUT=*number-of-seconds*

The number of seconds to wait for a socket to become readable before generating a communications timeout error.

Note: This variable is supplied for compatibility purposes only. IBM strongly encourages you to specify SOMDRECVWAIT instead of this variable. If SOMDRECVWAIT is not set and SOMDTIMEOUT is set, SOMDTIMEOUT will be the timeout value. If SOMDRECVWAIT is specified, then that value will be the timeout value, whether or not SOMDTIMEOUT is set.

Usage: Optional

Scope: Both

Default: 30

[SOMD_TCPIP] Stanza (TCP/IP Protocol)

This stanza, with the [somd] stanza, specifies settings for the TCP/IP protocol.

HOSTNAME=*mvs.sys.com*

The name by which the machine is known for this protocol. HOSTNAME can be set only in configuration files used by the SOM subsystem.

Usage: Required

Scope: System

Default: No default.

SOMDPORT=9393

The port address for the SOM subsystem's well-known address. When you use distributed SOMobjects, client programs establish connections with server programs through a daemon that runs in the SOM subsystem. The daemon listens for client requests on a well-known port.

Usage: Optional

Scope: System

Default: 9393

CSFactoryClass=SOMDCallStrmIOP::CallStreamFactoryIOP

The CallStreamFactory class name for this protocol.

Note: Always specify this exact value.

Usage: Required

Scope: System

Default: No default.

CSRegistrarClass=SOMDCSRegRI::CallStreamRegistrarRI

The CallStreamRegistrar class name for this protocol.

Note: Always specify this exact value.

Usage: Required

Scope: System

Default: No default.

CSTransportClass=SOMDtip::TCPIPTransportFactory

The Listening transport class name for this protocol.

Note: Always specify this exact value.

Usage: Required

Scope: System

Default: No default.

CSLocationName=SOMD_TCPIP

The name of the protocol the server uses to communicate with the location service (SOMDD).

Note: Always specify this exact value.

Usage: Required

Scope: System

Default: No default.

CSProfileTag=0;

The CallStreamFactory tag for this protocol. This is a protocol-unique tag used to represent the protocol within object references.

Note: Always specify this exact value.

Usage: Required

Scope: System

Default: No default.

CALL_POOL_SIZE=*size*

For a SOM process that is using the SOM distribution framework, the average number of outstanding remote calls that the SOM process expects to have simultaneously over this protocol.

Note: Always specify this exact value.

Usage: Required

Scope: Both

Default: 16

ENCAP_POOL_SIZE=*size*

The average number of requests that a client expects to send simultaneously over this protocol, or if a server, the average number of simultaneous objects the server expects to export (or return) over this protocol.

Default: 16

Usage: Optional

Scope: System

Default: 4

[somir] Stanza (Interface Repository)

This stanza specifies the settings for the SOMObjects Interface Repository.

SOMIR=*name1;name2;...namen*

The name (or list of names) of the files that contain the Interface Repository. The SOMIR keyword variable can reference an ordered list of separate IR files, which process from left to right. Taken as a whole, this gives the appearance of a single, logical interface repository. When running the SOMObjects compiler using the IR emitter, the rightmost data set in the list will be updated. Each

data set specification can be either an MVS data set or a file in the hierarchical file system.

When specifying *name* for the SOMIR variable, the following rules apply:

- To specify a fully-qualified MVS data set name, specify // before the name and put quotes around the name, such as:

```
//'SOMMVS.DSNAME'
```

- To use the current userid (or prefix) as the high-level qualifier, specify // before the name and omit the quotes, such as:

```
//DSNAME
```

- For an HFS file, omit // and specify complete path, such as:

```
/u/user1/filename
```

Examples:

To use the IBM-supplied data sets, specify:

```
SOMIR=//'SOMMVS.SGOSIR';//'SOMMVS.SGOSIRSM';
```

To use an HFS file named `localir`, then the IBM-supplied data sets, specify

```
SOMIR=/u/user1/localir;//'SOMMVS.SGOSIR';//'SOMMVS.SGOSIRSM';
```

Usage: Required

Scope: Both

Default: No default.

[somk] Stanza (SOMobjects Kernel)

This stanza specifies the settings for the SOMobjects kernel.

SMLANG=ENUS

The language used for SOMobjects messages. Valid values are ENUS for English and JAPN for Japanese.

Usage: Optional

Scope: System

Default: ENUS

SOMBASE=//SOMMVS

The high-level qualifier for SOMobjects-related data sets. If you do not want your data sets to have the high-level qualifier SOMMVS, change the variable, but leave the //.

Usage: Optional

Scope: System

Default: //SOMMVS

[somm] Stanza (Naming Service)

This stanza specifies the settings for the SOMobjects Naming Service.

GLOBAL_OBJREF_FILE=name

The name of an MVS data set or HFS file used by the SOMobjects configuration utility (SOM@CFG) to hold the global object reference.

On an install host, the object reference is written to this file.

On a DSOM host, the object reference is read from this file.

Example 1 (MVS data set name):

```
GLOBAL_OBJREF_FILE=//SOMNM.OBJREF
```

Example 2 (fully-qualified MVS data set name):

```
GLOBAL_OBJREF_FILE=/'SOMMVS.SOMNM.OBJREF'
```

Example 3 (HFS file name):

```
GLOBAL_OBJREF_FILE=/u/sommvs/somobjref.data
```

Usage: Optional

Scope: System

Default: *sommdir.SOMNM.REF*

SOMMNOBJREF=*object-reference-string*

The object reference for the local root of the naming tree. This variable is created and set by the configuration utility (SOM@CFG) and should not be modified.

Usage: Required

Scope: System

Default: No default

HOSTKIND=INSTALL | DSOM

The install type of this machine. This variable is created and set by the SOMobjects configuration utility (SOM@CFG) and should not be modified.

Usage: Required

Scope: System

Default: No default.

NAMING_SERVER_ALIAS=*alias*

The alias (32 character maximum) of the naming server.

Note: This specification must match the alias of the naming server specified

Usage: Optional

Scope: System

Default: NAMINGSERVER

[somras] Stanza (Error Log and Trace Facility)

This stanza specifies the settings for the SOMobjects error log and trace facilities.

SOMErrorLogFile=*name*

The name of the file or MVS data set where error log entries will be stored.

IBM recommends that all processes on a system share one error log file. Each data set specification can be either an MVS data set or a file in the hierarchical file system.

When specifying *name* for the SOMErrorLogFile variable, the following rules apply:

- To specify a fully-qualified MVS data set name, specify // before the name and put quotes around the name, such as:

```
//'SOMMVS.SOMERROR.LOG'
```

- To use the current userid (or prefix) as the high-level qualifier, specify // before the name and omit the quotes, such as:

```
//SOMERROR.LOG
```

- For an HFS file, omit // and specify complete path, such as:

```
/u/sommvs/somerror.log
```

Usage: Optional

Scope: Local

Default: somerror.log

Note: If the default is taken, the specification is treated as an MVS data set named SOMMVS.SOMERROR.LOG

SOMErrorLogSize=*size*

The size, in kilobytes, of the error log file. The default allows space for several hundred average-sized log entries.

Usage: Optional

Scope: Local

Default: 128

SOMErrorLogControl=*filter*

A filter to control what types of log entries will be included in the error log file. Multiple values may be specified, delimited by spaces. Valid values include INFO, WARNING, ERROR, and MAPPED_EXCEPTION.

Usage: Optional

Scope: Local

Default: WARNING ERROR MAPPED_EXCEPTION

SOMErrorLogDisplayMsgs=NO | YES

Indicates whether or not error log messages should be displayed to the standard output device in addition to the error log specified in the SOMErrorLogFile. IBM recommends that you specify YES during initial installation and when trying to debug a problem. At other times, specify NO.

Usage: Optional

Scope: Local

Default: YES

SOM_TraceLevel=*level*

This variable will establish the SOM trace level for the SOM subsystem only. (For other processes, the SOM_TraceLevel value is obtained through an OpenEdition environment variable setting.)

Valid values are:

0 Turns SOM tracing off.

1 Minimum trace data captured: error messages and exceptions.

- 2 Capture above, plus communication messages.
See the description of the SOMDCOMMDEBUGFLAG environment variable.
- 3 Capture above, plus entry/exit and calls to external functions.
- 4 Capture above, plus miscellaneous messages.
- 5 Capture above, plus somras messages.

Usage: Optional

Scope: Used for the SOM subsystem only.

Default: 0

MVSTraceLog=*hlq*

The high-level qualifier of the trace data set. This variable must be specified to get trace records written to a data set. If SOM_TraceLevel is greater than zero and MVSTraceLog is specified, a SOMobjects trace data set is allocated for each process (address space) during initialization of that process. If you specify a value for MVSTraceLog, IBM recommends specifying MVSTraceLog=SOMMVS. The name of the SOMobjects trace data set is a unique name generated by SOMobjects. The name is reported to the user in message GOS20007I.

Usage: Optional

Scope: Local

Default: No default. If this variable is not specified, tracing will be done, but only to an in-memory trace buffer.

MVSTraceLogSize=*nnn sss*

The size of the SOMobjects trace data set, where:

nnn is a decimal value.

sss is either TRK or CYL, for tracks and cylinders, respectively.

Usage: Optional

Scope: Local

Default: 100 times the value of SOM_TraceLevel, in units of TRK (tracks).

[somsec] Stanza (Security Service)

This stanza specifies the settings for the SOMobjects security service.

Note: If RACF PassTicket support is used, you need only specify the DISABLE_AUTHN variable. In this case, the target application server does not need to be running on the OS/390 platform.

USER=*userid*

The user ID used to authenticate a distributed SOMobjects client. If the USER variable is not set, the userid associated with the address space under which the client is executing will be used. Note that the system on which the security server runs must have a definition (a RACF user profile) for this user ID.

Never specify the USER keyword in the SOM subsystem's configuration file. Since the SOM subsystem's configuration file values are global, the user ID would be exposed to all SOMobjects applications.

Usage: Optional

Scope: Local

Default: If you do not specify the USER keyword, it will default to the user ID associated with the address space in which the SOMobjects application is executing.

PASSWD=*password*

The password used to authenticate a distributed SOMobjects client. Corresponds to the user ID that identifies the user of the SOMobjects application. Note that the system on which the security server runs must have a definition for this user ID/password pair. For a SOMobjects application communicating only with OS/390 servers, PASSWD may be omitted if RACF PassTicket support is enabled. PassTickets offer greater security. For information on enabling PassTicket support, see “Setting up the PassTicket Environment” on page 72.

Never specify the PASSWD keyword in the SOM subsystem's configuration file. Since the SOM subsystem's configuration file values are global, the password would be exposed to all SOMobjects applications. If PassTicket support is not available, you are establishing a connection with a non-OS/390 server, the best strategy is to specify an appropriate PASSWD value in the local configuration file of each SOMobjects application.

Usage: Optional

Scope: Local

Default: Null

DISABLE_AUTHN=TRUE | FALSE

Set DISABLE_AUTHN to TRUE to turn off authentication. Authentication is enabled by default.

Note: A secure server rejects requests from unauthenticated clients.

Usage: Optional

Scope: Local

Default: TRUE

SECURITY_SERVER_ALIAS=*alias*

The alias (32 character maximum) of the security server.

Usage: Optional

Scope: System

Default: SECURITYSERVER

[SOM_POSSOM] Stanza (Persistent Object Service)

This stanza specifies the settings for the persistent object service environment.

POS_POMDATA=*name*

The path and file name of the POM (persistent object manager) data file. Each data set specification can be either an MVS data set or a file in the hierarchical file system.

When specifying *name* for the SOM_POSSOM variable, the following rules apply:

- To specify a fully-qualified MVS data set name, specify // before the name and put quotes around the name, such as:

```
//'SOMMVS.SGOSMISC(GOSPMDAT)'
```

- To use the current userid (or prefix) as the high-level qualifier, specify // before the name and omit the quotes, such as:

```
//SGOSMISC(GOSPMDAT)
```

- For an HFS file, omit // and specify complete path, such as:

```
/u/sommvs/gospmdat
```

Usage: Optional

Scope: Both

Default: //'SOMMVS.SGOSMISC(GOSPMDAT)'

POSIX_StreamCreation=LOCAL | NONLOCAL

The location of the stream for the POSIX protocol. To create the stream in the same process as the POM (persistent object manager), specify NONLOCAL. The persistent object manager is a class within the SOM persistence framework.

To create the stream in the same process as the PDS (persistent data store), specify LOCAL. The persistent data store is a class within the SOM persistence framework.

Usage: Optional

Scope: Both

Default: NONLOCAL

BTREE_StreamCreation=LOCAL | NONLOCAL

The location of the stream for the BTREE protocol. To create the stream in the same process as the POM, specify NONLOCAL. To create the stream in the same process as the PDS, specify LOCAL.

Usage: Optional

Scope: Both

Default: NONLOCAL

Chapter 6. SOMobjects Coexistence Considerations

If you are installing SOM on more than one system, read one of the following sections:

- “Data Sets to Put on Shared Volumes” if the systems are not part of a sysplex.
- “Sysplex-specific Coexistence Considerations” on page 92 if the systems are all within a sysplex.

If you have a sysplex and one or more systems that are not part of the sysplex, read “Sysplex-specific Coexistence Considerations” on page 92 for the considerations among the members of the sysplex, and read “Data Sets to Put on Shared Volumes” for the non-sysplex systems.

Multisystem Coexistence Considerations

Data Sets to Put on Shared Volumes

The following data should be on shared volumes in a SOMobjects multisystem environment:

- SOMobjects code
- The Interface Repository
- The global configuration file (SOMMVS.SOMENV.INI)
- Binary databases are allocated as key-sequenced VSAM clusters under the *sommdir* high-level qualifier, such as:

```
sommdir.ALIAS.DB  
sommdir.ALIASDAT.DB  
sommdir.IMPL.DB  
sommdir.OBJ.DB  
sommdir.nnnnnnnn.BND
```

If cluster allocation is under SMS control, then ensure that storage administration tables provide shared volume allocation. Otherwise, specify shared volumes on the VOLUME variable of the [sombt] stanza in the configuration file. See “[sombt] Stanza (BTree Framework)” on page 77.

SOM Subsystem Considerations

The following are SOM subsystem considerations when configuring for a multi-system environment:

- One procedure can be used to start the SOM subsystems if &SYSNAME is used as part of the configuration file name.

The following example shows how a single procedure could be used to start the SOM subsystems by using &SYSNAME as part of the configuration file name. The configuration file that has &SYSNAME in it would contain only the HOSTNAME setting, because HOSTNAME is different for each system on which the SOM subsystem is started.

```
//SOMENV DD DSN=SOMMVS.HOSTNAME.INI.&SYSNAME,DISP=SHR  
// DD DSN=SOMMVS.HOSTNAME.INI,DISP=SHR
```

- The first SOMENV DD card above identifies a configuration file that contains the system TCPIP hostname as specified with the HOSTNAME environmental variable in the [somed] and [SOMD_TCPIP] stanzas. There must be one of these configuration files for each system on which the SOM subsystem runs.
- The second DD card concatenated to the SOMENV DD above specifies the global configuration file that contains the common stanza information to be used across similar systems.
- If multiple procedures are used to start the SOM subsystems, specify JOBNAME so that they are all assigned the same subsystem name. For example, if you start procedures SOM1 on SYS1 and SOM2 on SYS2, your SOM subsystems on those two systems would appear as SOM1 and SOM2, respectively. You can specify the JOBNAME keyword on the MVS START commands as follows.

```
START SOM1,JOBNAME=SOM
START SOM2,JOBNAME=SOM
```

so that the SOM subsystem is called SOM on both systems.

SOMobjects Server Considerations

- If a server procedure specifies a configuration file on the SOMENV DD, it must not contain HOSTNAME.

SOMobjects Configuration Utility (SOM@CFG) Considerations

- Run SOM@CFG on each system on which SOMobjects is to run. Use the -i option for the first system and the -d option for subsequent systems.

RACF Considerations

- The RACF profiles must specify the various combinations of *SOM.subsystem-name* that are desired for any one system.

For more information see Chapter 4, “Configuring and Administering Security for SOMobjects” on page 67.

Sysplex-specific Coexistence Considerations

Kind of Sysplex Required

To run SOM@CFG or distributed applications, the system must be a monoplex or a multisystem sysplex.

Data Sets to Put on Shared Volumes

The following data should be on shared volumes in a SOMobjects sysplex environment:

- SOMobjects code
- The Interface Repository
- The global configuration file (SOMMVS.SOMENV.INI)
- Binary databases are allocated as key-sequenced VSAM clusters under the *somdir* high-level qualifier, such as:

```
somdir.ALIAS.DB
smdir.ALIASDAT.DB
smdir.IMPL.DB
smdir.OBJ.DB
smdir.nnnnnnnn.BND
```

If cluster allocation is under SMS control, then ensure that storage administration tables provide shared volume allocation. Otherwise, specify shared volumes on the VOLUME variable of the [sombt] stanza in the configuration file. See “[sombt] Stanza (BTree Framework)” on page 77.

SOM Subsystem Considerations

The following are SOM subsystem considerations when configuring for a sysplex environment:

- All members of a sysplex that are to run SOMobjects must have the SOM subsystem running. Also, if that member is to participate in workload balancing, the member must be in goal mode.

One procedure can be used to start the SOM subsystems if &SYSNAME is used as part of the configuration file name.

The following example shows how a single procedure could be used to start the SOM subsystems by using &SYSNAME as part of the configuration file name. The configuration file that has &SYSNAME in it would contain only the HOSTNAME setting, because HOSTNAME is different for each system in the sysplex where the SOM subsystem is started.

```
//SOMENV DD DSN=SOMMVS.HOSTNAME.INI.&SYSNAME,DISP=SHR
//      DD DSN=SOMMVS.HOSTNAME.INI,DISP=SHR
```

- The first SOMENV DD card above identifies a configuration file that contains the system TCPIP hostname as specified with the HOSTNAME environmental variable in the [somb] and [SOMD_TCPIP] stanzas. There must be one of these configuration files for each system in the sysplex.
 - The second DD card concatenated to the SOMENV DD above specifies the global configuration file that contains the common stanza information to be used in the sysplex.
- If multiple procedures are used to start the SOM subsystems, specify JOBNAME so that they are all assigned the same subsystem name. For example, if you start procedures SOM1 on SYS1 and SOM2 on SYS2, your SOM subsystems on those two systems would appear as SOM1 and SOM2, respectively. You can specify the JOBNAME keyword on the MVS START commands as follows.

```
START SOM1,JOBNAME=SOM
START SOM2,JOBNAME=SOM
```

so that the SOM subsystems are called SOM on both systems.

SOMobjects Server Considerations

- If a server procedure specifies a configuration file on the SOMENV DD, it must not contain HOSTNAME.

SOMobjects Configuration Utility (SOM@CFG) Considerations

- SOM@CFG only needs to be run once for the sysplex, with the -i option (as specified in the IBM-supplied example).
- Do not start other SOM subsystems until after SOM@CFG has been run in the first member.

RACF Considerations

- The RACF profiles must specify the various combinations of *SOM.subsystem-name* that are desired for any one particular member of the sysplex.

For more information see Chapter 4, “Configuring and Administering Security for SOMobjects” on page 67.

Chapter 7. Preparing the OpenEdition Environment to Run SOMobjects Utilities

SOMobjects provides several utilities, which can be run in a TSO/E environment or in the OpenEdition shell:

- The **IRDUMP utility**, which prints the contents of the SOMobjects interface repository (IR). For details, see the section on accessing objects in the interface repository in the *OS/390 V2R4.0 SOMobjects Messages, Codes, and Diagnosis*.
- The **REGIMPL utility**, which registers a server implementation and the classes it supports in the SOMobjects implementation repository. For details, see “Using the REGIMPL Utility to Register Servers” on page 53.
- The **SOMDSVR server program**, which is described in the section on controlling a server application in the runtime environment in *OS/390 V2R4.0 SOMobjects Programmer's Guide*.
- The **SOMobjects compiler**, which translates IDL definitions of SOMobjects classes into a set of bindings for specific languages. For details, see *OS/390 V2R4.0 SOMobjects Programmer's Guide*.
- The **object services server**, which sets up persistent object references and manages object metastates. For details, see *OS/390 V2R4.0 SOMobjects Programmer's Reference, Volume 2*.

If you want to run SOMobjects utilities in the OpenEdition shell, take the following steps:

- “Creating Dummy Files for SOMobjects Functions”
- “Setting Up the STEPLIB Concatenation for SOMobjects” on page 97

Creating Dummy Files for SOMobjects Functions

Create a “dummy” file for each utility in the default path for executable programs (/bin).

To edit files in the (/bin) directory, you must have *superuser* authority to the OpenEdition shell. A *superuser* has special authority to perform certain tasks (such as mounting and unmounting a file system) and can access all OpenEdition services and the files in the hierarchical file system (HFS). For information about how to obtain *superuser* authority, see the section on switching to superuser in *OS/390 OpenEdition User's Guide*.

Use the following procedure to create the desired “dummy” files:

1. From TSO/E, enter the OMVS command to enter OpenEdition:

```
OMVS
```

2. Enter the cd command to change the working directory to /bin:

```
cd /bin
```

3. Enter the echo command to create hierarchical file system (HFS) files for each function as follows:

For the IRDUMP utility:

```
echo irdump > irdump
```

For the REGIMPL utility:

```
echo regimpl > regimpl
```

For the SOMobjects compiler:

```
echo sc > sc
```

For the SOMobjects generic server program (somsdsvr):

```
echo somdsvr > somdsvr
```

For the SOMobjects objects services server program (somossvr):

```
echo somossvr > somossvr
```

4. Enter the chmod command to allow the actual command to reside in a data set that is part of a STEPLIB or link list. The chmod command also sets a “sticky bit,” which permits files under the /bin directory to be deleted and renamed only by a superuser. The sticky bit also causes the system, when processing an exec request of the command, to look through the standard system search order for a member with the command name. Enter chmod for each requested function.

The following example shows how to enter chmod for the SOMobjects compiler:

```
chmod -R 1755 irdump
chmod -R 1755 regimpl
chmod -R 1755 sc
chmod -R 1755 somdsvr
chmod -R 1755 somossvr
```

5. Enter the ls command to display information about the created files. Enter ls for each of the requested functions.

The following example shows how to enter ls for the SOMobjects compiler:

```
ls -Fl a sc
```

The output from the command is as follows:

```
-rwxr-xr-t  1 AHLEE  TASKS          3 Dec  5 14:58 sc*
```

Make sure that the permissions are as shown, and that an asterisk (*) follows the file name (the asterisk indicates that the file is executable).

Setting Up the STEPLIB Concatenation for SOMobjects

To add SOMobjects code to STEPLIB for an OpenEdition session, take the following steps.

Note: IBM recommends that you place the listed commands in a shell script, since you need to enter them each time the shell is invoked.

Step 1. Export the SOMobjects executables:

```
export STEPLIB="SOMMVS.SGOSLOAD"
export STEPLIB="$STEPLIB:CEE.SCEERUN"
export STEPLIB="$STEPLIB:CBC.SCBCMP"
export STEPLIB="$STEPLIB:CBC.SCLBDLL"
```

The specified data sets must be cataloged OS/390 load libraries to which you have at least READ access authority.

Step 2. Rerun the shell command in the current address space, then export the current STEPLIB:

```
exec sh -L
export STEPLIB=CURRENT
```

Chapter 8. Initializing somOS::Server-based Application Servers

To initialize a somOS::server-based application server, take the following steps:

1. Select one of the following server programs to run:
 - a. Select SOMDSVR unless one of the following considerations applies.
 - b. Select SOMOSSVR if your application uses any of the object services (see *OS/390 V2R4.0 SOMobjects Object Services*.)
 - c. If you have unique requirements not met by either of the IBM-supplied server programs, select a customer-written server program. For information on writing a server program, see *OS/390 V2R4.0 SOMobjects Programmer's Guide*.
2. Configure the application server as described in "Configuring and Running an Application Server" on page 10.

Hints & Tips

If you chose SOMOSSVR, a one-time initialization step is required for the application server:

```
somossvr -i -a server-alias
```

Starting the server with the `-i` option instructs the server to initialize its own VSAM databases. These will be VSAM clusters defined under the high-level qualifier specified by the `SOMDDIR` environment variable that is in effect for the server.

Chapter 9. Setting up the ISPF Interface to the SOMobjects Compiler

The ISPF panel interface to the SOMobjects compiler provides an interactive dialog in which users specify compiler parameters. You can use the panels as a stand-alone program or as an option from the ISPF Batch Selection panel.

To set up this ISPF panel interface, allocate the ISPF dialog elements as follows:

- ISPF panels: Allocate data set SOMMVS.SGOSPNLS to ISPLLIB.
- REXX execs: Allocate data set SOMMVS.SGOSREXX to SYSEXEC.
- ISPF messages: Allocate data set SOMMVS.SGOSMSGs to ISPLLIB.
- ISPF skeletons: Allocate data set SOMMVS.SGOSSKEL to ISPLLIB.

The SOMobjects compiler dialog includes online help panels and a brief tutorial.

Setting Up How the Panels Will Be Invoked

Select the method to invoke the SOMobjects compiler panel interface through the GOSRX001 member of the SOMMVS.SGOSREXX data set. Within the GOSRX001 member, set the return value to:

- **N** to invoke the standalone panel interface
- **Y** to invoke the panel interface through the ISPF Batch Selection panel

Standalone Considerations

If you use the the SOMobjects compiler ISPF panel interface as a standalone program, you can display and edit the JCL generated from the panel.

If you invoke the panel interface through the ISPF Batch Selection panel, you cannot display and edit the JCL.

After installing, activating and modifying the JCL skeletons, call the standalone program with the command TSO GOSRMAIN from an ISPF panel.

Modify the Skeletons for Background Compilation

To support background compilation, you must change the skeleton JCL in SOMMVS.SGOSSKEL(GOSPSOM2). You can specify any job card information that your installation would normally use.

```
//&ZUSER.F JOB '?','?', '&ZUSER', CLASS=U,  
// NOTIFY=????????, MSGCLASS=H, MSGLEVEL=(1,1)  
)ENDSEL  
// SET SOM=SOMMVS  
// SET LE=CEE
```

If you have changed the default names of the SOMobjects compiler elements, you may need to change the stub JCL in SOMMVS.SGOSSKEL(GOSPSTUB).

1. First, make any necessary changes to the four SET statements shown below. These are:

```
)SET SOMPGM = SC
)SET TEMPDD = SOMPARMS
)SET REXXLIB = SOMMVS.SGOSREXX
)SET REXXPGM = GOSRSOMP
```

where:

- SOMPGM is the SOMobjects compiler command
 - SOMPARMS is a temporary data set to hold compiler parameters
 - REXXLIB is the REXX library that holds the SOMobjects compiler panel program's REXX code
 - REXXPGM is the name of the REXX program shipped as GOSRSOMP
2. You also might need to change the data set names of the standard ISPF libraries shown below:

```
//ISPMLIB !DD DISP=SHR,DSN=SYS1.ISP.ISPMENU
//          !DD DISP=SHR,DSN=SYS1.ISR.ISRMENU
//ISPPLIB !DD DISP=SHR,DSN=SYS1.ISP.ISPPENU
//          !DD DISP=SHR,DSN=SYS1.ISR.ISRPENU
//ISPSLIB !DD DISP=SHR,DSN=SYS1.ISR.ISRSENU
//ISPTLIB !DD DISP=SHR,DSN=ISP.ISPF.TABLES
```

Chapter 10. Configuring a DSOM Host

The SOMobjects configuration described in Chapter 1, “Configuring Your SOMobjects Environment” on page 1 is for a SOMobjects “install host.” An install host is the first system on which you install SOMobjects. It has both the global SOMobjects naming server and SOMobjects security server.

There is one global SOMobjects naming server and SOMobjects security server in a SOMobjects domain.

Additional systems that you configure into an existing SOMobjects domain are called DSOM hosts. A DSOM host has its own local SOMobjects naming server. Through the local SOMobjects naming server, SOMobjects applications running on a DSOM host can access the global SOMobjects naming server. A DSOM host does not have a SOMobjects security server.

In a sysplex, only one system can be configured as an install host or a DSOM host. For additional sysplex considerations, see “Sysplex-specific Coexistence Considerations” on page 92.

Configuring a DSOM host is similar to installing an install host. Take the following steps:

- ___ 1. Complete the configuration steps specified in “Getting Started with Non-Distributed SOMobjects” on page 1.
- ___ 2. Create a JCL procedure for SOMobjects naming server.
- ___ 3. Define a security identity and authorizations for SOMobjects naming server.
- ___ 4. Define SOMobjects naming server to WLM.
- ___ 5. Enable WLM server management.
- ___ 6. Run the SOMobjects configuration utility (SOM@CFG).
- ___ 7. Make your naming server secure with the REGIMPL utility.
- ___ 8. Recycle the SOM subsystem.

DSOM Host: Creating a JCL Procedure for the SOMobjects Naming Server

The distributed SOMobjects environment requires one server address space for the SOMobjects naming server. This server runs as a started task. It requires a JCL procedure, and is started automatically by the SOM subsystem. Create a JCL procedure for it by copying sample procedure SOMMVS.SGOSJCL(GOSSOSSVR) to SYS1.PROCLIB as member GOSSOSSVR.

DSOM Host: Defining Security Identity and Authorizations for the SOMobjects Naming Server

The SOMobjects naming server requires a user identity for authorization purposes. Assign a user identity to the GOSSOSSVR procedure and authorize it to establish the naming server by issuing the following RACF commands:

```

RDEFINE STARTED GOSOSSVR.* STDATA(USER(userid) +
      GROUP(group-name))
SETROPTS RACLIST(STARTED) REFRESH
SETROPTS CLASSACT(SERVER)
SETROPTS RACLIST(SERVER)
RDEFINE SERVER SOM.SOM.NAMINGSERVER UACC(NONE)
PERMIT SOM.SOM.NAMINGSERVER CLASS(SERVER) ID(userid) ACCESS(READ)
SETROPTS RACLIST(SERVER) REFRESH

```

The variable *userid* is the user name, and *group-name* is an appropriate RACF group of which the server should be a member, such as SYS1.

Note: The user identified by *userid* must also be authorized to OpenEdition.

For additional information on defining security identity and authorizations for the SOMObjects naming server, see Chapter 4, “Configuring and Administering Security for SOMObjects” on page 67.

Hints & Tips

The SOMObjects Naming server is a non-secure server by default. This means clients are free to establish binds to the SOMObjects Naming server without authorization. Bind authorization is controlled through the RACF CBIND class.

Secure this server by changing its SOMObjects registration using the REGIMPL utility. Choose option 3 (Change) and answer 'y' to the prompt "Make server secure?". For more information on changing a server's registration, see “Changing a Server in the Implementation Repository” on page 56.

You must authorize clients to bind to secure servers by setting up profiles in the RACF CBIND general resource class. For additional information on authorizing clients to servers, see Chapter 4, “Configuring and Administering Security for SOMObjects” on page 67.

DSOM Host: Registering the SOMObjects Naming Server with WLM

If a requested server is not already running, it will automatically be started by WLM when required by a SOMObjects client program. The SOMObjects naming server must be defined to WLM. Use the IWMARIN0 TSO command to establish these definitions:

- ___ 1. Issue TSO IWMARIN0
- ___ 2. Select option **2**. Extract the definition from the WLM couple data set
 Selecting option **2** assumes you already have an active WLM definition in use on your system. (If this does not describe your situation, refer to Chapter 2, “Configuring and Administering SOMObjects with Workload Manager (WLM)” on page 15 for further information.)
- ___ 3. Select option **9** *Application Environments*

The following is the screen for the application environment setup for the naming server.

```

Application-Environment  Notes  Options  Help
-----
                          Create an Application Environment
Command ==> _____
Application Environment . . . NAMINGSERVER
Description . . . . . SOM Naming Server
Subsystem Type . . . . . SOM
Procedure Name . . . . . GOSOSSVR
Start Parameters . . . . . PARM='-a NAMINGSERVER'
Limit on starting server address spaces for a subsystem instance:
3  1. No limit
   2. Single address space per system
   3. Single address space per sysplex

```

DSOM Host: Enabling WLM Server Management

The SOMObjects naming server can be started only if WLM has been enabled to perform server management. Enable WLM server management by installing the definition to the couple data set, activating the service policy, and entering WLM goal mode.

Note: IBM recommends that you set up performance goals for the server. Unless performance goals are set up for the server, the server might receive a lower proportion of system resources than what might be expected.

See Chapter 2, “Configuring and Administering SOMObjects with Workload Manager (WLM)” on page 15 for information on performing these tasks.

DSOM Host: Running the SOMObjects Configuration Utility (SOM@CFG)

The SOMObjects configuration utility (SOM@CFG) runs as a batch job that creates control files used by various SOMObjects facilities. These control files require 50 tracks of available DASD.

To run the SOMObjects configuration utility, take the following steps:

1. Copy the SOMMVS.SOMNM.REF file from your install host to the system you are installing as a DSOM host.

Note: This file was created by running SOM@CFG on your install host. SOMMVS.SOMNM.REF is the default name for this file. If you set the GLOBAL_OBJREF_FILE keyword on the [somnm] stanza in your configuration file, copy the file whose name is specified on that keyword.

2. Ensure that the SOM@CFG job is submitted by a user ID that has the following authorizations:
 - Write access to SOMMVS.* data sets
 - OpenEdition access
3. Edit member GOSCFG in data set SOMMVS.SGOSJCL as follows:
4. Add job card information.
5. Change the -i parameter on SOM@CFG to -d. Follow instructions in SOMMVS.SGOSJCL(GOSCFG).

6. Submit the job.

Hints & Tips

Error Log Entries that can be Ignored

SOM@CFG creates error log entries that can be ignored if SOM@CFG completes successfully. You can expect log entries with the following error codes:

- Error code is 30046 [SOMDERROR_EntryNotFound]. SOM@CFG registers the Naming Service and Security Service in the Implementation Repository. When querying whether entries for these servers already exist, this error may be raised.
- Error code is 30088 [SOMDERROR_NamingNotActive]. When registering with the Implementation Repository, SOMobjects will attempt to store the information in the Naming Service. If the naming server is not running, this error will be raised.

Sysplex Considerations:

- Before running SOM@CFG in a sysplex, determine whether the VSAM clusters created by SOM@CFG need to be on DASD that is shared across the sysplex. For more information, see “Data Sets to Put on Shared Volumes” on page 91.
- SOM@CFG needs to be run only once for the entire sysplex, using the `-i` option (as specified in the IBM-supplied example).

DSOM Host: Recycling the SOM Subsystem

To complete the distributed configuration, the SOM subsystem must be re-cycled. Recycling the SOM subsystem is required so that it can reconfigure itself to incorporate data created by the SOMobjects configuration utility.

Re-cycle the SOM subsystem by issuing the following operator commands:

1. STOP SOM
2. START SOM

Note: The SOMobjects naming server will terminate when the SOM subsystem is recycled. This is expected. After the SOM subsystem is recycled, the server will start on demand when client requests are initiated.

DSOM Host: Capacity Planning for Running Many Distributed Clients

As the number of clients being run increases, the number of socket connection requests to the daemon and servers which they access also increases. The rate at which the daemon and servers are able to accept client connect requests may decrease to the point where the backlog queue of socket connect requests waiting for an accept is exceeded. When this happens, the client receives a communication error exception resulting from a timeout received while attempting the connection request. To circumvent this, your installation might need to increase the SOMAXCONN configuration parameter in your TCPIP.PROFILE data set. The value should be increased to the greater of:

- The number of servers you anticipate will be connecting to the daemon simultaneously.
- The number of clients you anticipate will be connecting to your SOMobjects servers simultaneously.

You may also find as you increase the number of clients running against your servers that additional system tuning may be required. If unexpected exceptions (such as communication error exceptions) begin to occur you should also examine runtime diagnostic messages that may be displayed by programs such as TCP/IP. For example, TCP/IP may display error messages under the SYSDEBUG ddname. Also, if the SNALINK LU6.2 interface is configured with TCP/IP, the SYSPRINT ddname may contain error messages displayed by SNA.

DSOM Host: Configuring and Running an Application Server

To test your DSOM host configuration, Configure and run an application server as described in “Configuring and Running an Application Server” on page 10.

Index

Special Characters

*

- adding to interface references
- SMADDSTAR keyword in [somc] stanza 80

A

activating a service policy in WLM 48

address spaces

- limit on starting server address spaces in a sysplex 24

alias

- NAMING_SERVER_ALIAS keyword in [somm] stanza 85
- SECURITY_SERVER_ALIAS keyword in [somsec] stanza 88
- specifying alias of naming server in WLM 18
- specifying alias of security server in WLM 23
- using REGIMPL to list aliases of servers 58

allocation

- allocating the implementation repository 53

APPL general resource class

- creating profiles 72

application servers

- configuring and running 10
- creating a JCL procedure (on install hosts) 10
- defining a security identity (on install hosts) 11
- defining to WLM 11
- initializing somOS::server-based application servers 99
- registering with the SOM subsystem 11

ASID (address space identifier)

- required for &sommmvs. 3

auditing 68

authentication

- description 72
- DISABLE_AUTHN keyword in [somsec] stanza 88

B

BTREE_StreamCreation keyword in

[SOMD_POSSOM] stanza 89

C

CALL_POOL_SIZE keyword in [SOMD_TCPIP]

stanza 83

CallStreamFactory class name

- CSFactoryClass keyword in [SOMD_TCPIP] stanza 82

CallStreamFactory tag

- CSProfileTag keyword in [SOMD_TCPIP] stanza 83

CallStreamRegistrar class name

- CSRegistrarClass keyword in [SOMD_TCPIP] stanza 82

capacity planning 12

capacity planning (on DSOM hosts) 106

CBIND general resource class

- creating profiles 71
- description 5

chmod command

- for SOMobjects utilities 96

class

- registering in the implementation repository 53
- using REGIMPL to add a class to a server 59
- using REGIMPL to add a class to all server 60
- using REGIMPL to delete a class from all servers 60
- using REGIMPL to delete a class from one servers 61
- using REGIMPL to list classes in the implementation repository 60

classifying work in WLM 41

client authentication

- description 72
- DISABLE_AUTHN keyword in [somsec] stanza 88
- non-secure servers 67
- secure servers 71

coexistence considerations 91

communication protocol

- SOMDPROTOCOLS keyword in [somed] stanza 81

communications debugging

- SOMDCOMMDEBUGFLAG keyword in [somed] stanza 80

compiler

- ISPF interface to the SOMobjects compiler 101

configuration utility (SOM@CFG)

- multisystem considerations 92
- running (on DSOM hosts) 105
- running (on install hosts) 9
- sysplex considerations 94

CSFactoryClass keyword in [SOMD_TCPIP]

stanza 82

CSLocationName keyword in [SOMD_TCPIP]

stanza 83

CSProfileTag keyword in [SOMD_TCPIP] stanza 83

CSRegistrarClass keyword in [SOMD_TCPIP]

stanza 82

CSTransportClass keyword in [SOMD_TCPIP]

stanza 82

D

data sets to put on shared volumes 91, 92

debugging

SOMDCOMMDEBUGFLAG keyword in [somed] stanza 80

DISABLE_AUTHN keyword in [somsec] stanza 88

displaying a server's status 51

DSOM host

configuring (on DSOM hosts) 103
specifying in global configuration file 85

E

echo command

for SOMobjects utilities 96

emitter framework utilities

NEWEMITEFW keyword in [somic] stanza 79

emitter prefix

SMEMITPREFIX keyword in [somic] stanza 79

emitters

SMEMIT keyword in [somic] stanza 79

ENCAP_POOL_SIZE keyword in [SOMD_TCPIP] stanza 83

error log

SOMErrorLogControl keyword in [somras] stanza 86
SOMErrorLogDisplayMsgs keyword in [somras] stanza 86
SOMErrorLogFile keyword in [somras] stanza 85
SOMErrorLogSize keyword in [somras] stanza 86

export command

 97

extended CSA (ECSA)

specifying sufficient resources 2

extended LPA (ELPA)

specifying sufficient resources 2

F

fast path for REGIMPL 62

G

global object reference

GLOBAL_OBJREF_FILE keyword in [somm] stanza 84

global object reference string

SOMNMOBJREF keyword in [somm] stanza 85

GLOBAL_OBJREF_FILE keyword in [somm] stanza 84

GSOMDOBJ general resource class

creating profiles 73

H

high-level qualifier for SOMobjects-related data sets

SOMBASE keyword in [somp] stanza 84

high-level qualifier for SOMobjects-related data sets
(continued)

SOMDDIR keyword in [somed] stanza 80

hops

MAXIMUMHOPS keyword in [somed] stanza 81

HOSTKIND keyword in [somm] stanza 85

HOSTNAME environment variable

coexistence considerations 91
sysplex considerations 93

HOSTNAME keyword

specifying for the SOM subsystem 4

HOSTNAME keyword in [SOMD_TCPIP] stanza 82

HOSTNAME keyword in [somed] stanza 81

I

i option for SOMOSSVR program 99

i option on SOM@CFG 92, 94

IEFSSNx parmlib member

specifying SOMobjects values 2

implementation repository

adding a class to a server implementation 59
adding a class to all servers 60
adding a server 54
allocation considerations 53
changing a server 56
deleting a class from all server implementations 60
deleting a server 61
deleting one class from a server implementation 61
description 53
listing aliases 58
listing classes in the implementation repository 60
registering classes and servers 53
saving your data and exiting REGIMPL 62
showing all servers 58
showing one server 57
SOMIR keyword in [somir] stanza 83

install host

configuring for distributed SOMobjects applications 7
configuring for non-distributed SOMobjects applications 1
description (on DSOM hosts) 103
specifying in global configuration file 85

ISPF interface to the SOMobjects compiler 101

L

language used for SOMobjects messages

SMLANG keyword in [somp] stanza 84

listening transport class name

CSTransportClass keyword in [SOMD_TCPIP] stanza 82

LNKLSTxx parmlib member

specifying SOMobjects values 2

location forwarding requests
MAXIMUMHOPS keyword in [somd] stanza 81

location service protocol name
CSLocationName keyword in [SOMD_TCPIP]
stanza 83

M

making a server available 51

managing servers using WLM commands 51

MAXIMUMHOPS keyword in [somd] stanza 81

method-level access control
setting up 73

migration
allocation of implementation repository 53

monoplex 92

multisystem sysplex 92

MVSTraceLog keyword in [somras] stanza 87

MVSTraceLogSize keyword in [somras] stanza 87

N

naming server
creating a JCL procedure (on DSOM hosts) 103
creating a JCL procedure (on install hosts) 7
defining a security identity (on DSOM hosts) 103
registering the naming server with WLM (on DSOM
hosts) 104
specifying alias of naming server in WLM 18

**NAMING_SERVER_ALIAS keyword in [sommn]
stanza** 85

NEWEMITEFW keyword in [somc] stanza 79

non-secure servers
client authentication 67
security considerations 67

O

OpenEdition environment
preparing for SOMobjects utilities 95

outstanding remote calls
CALL_POOL_SIZE keyword in [SOMD_TCPIP]
stanza 83

P

parmlib members
specifying SOMobjects values 2

PassTickets
considerations if not used 67
setting up 72

PASSWD keyword in [somsec] stanza 88

PDS (persistent data store)
POSIX_StreamCreation keyword in
[SOMD_POSSOM] stanza 89

performance
managing with WLM 30

persistent data store (PDS)
POSIX_StreamCreation keyword in
[SOMD_POSSOM] stanza 89

persistent object manager (POM)
POS_POMDATA keyword in [SOMD_POSSOM]
stanza 88
POSIX_StreamCreation keyword in
[SOMD_POSSOM] stanza 89

POM (persistent object manager)
POS_POMDATA keyword in [SOMD_POSSOM]
stanza 88
POSIX_StreamCreation keyword in
[SOMD_POSSOM] stanza 89

pool size
CALL_POOL_SIZE keyword in [SOMD_TCPIP]
stanza 83
ENCAP_POOL_SIZE keyword in [SOMD_TCPIP]
stanza 83

port address of SOM subsystem
SOMDPORT keyword in [SOMD_TCPIP] stanza 82

**POS_POMDATA keyword in [SOMD_POSSOM]
stanza** 88

**POSIX_StreamCreation keyword in
[SOMD_POSSOM] stanza** 89

principal 68

printing of processing messages
SMNOPRINTNAME keyword in [somc] stanza 79

PROGxx parmlib member
specifying SOMobjects values 2

prompting mode of the REGIMPL command 54

protocol
SOMDPROTOCOLS keyword in [somd] stanza 81

R

RACF
multisystem considerations 92
sysplex considerations 94

RACF (Resource Access Control Facility)
support for SOMobjects 68

RACF groups
using to simplify RACF administration 68

RACF PassTickets
considerations if not used 67
setting up 72

REGIMPL help screen 62

REGIMPL utility 53
s option 71
saving your data and exiting REGIMPL 62

**registering classes in the implementation
repository** 53

**registering servers in the implementation
repository** 53

S

s option on REGIMPL command 71

secure mode

configuring a server to run in secure mode 71

secure servers

authorizing clients 71

configuring a server to run in secure mode 71

security

multisystem considerations 92

setting up security for SOMobjects 67

sysplex considerations 94

security considerations for non-secure servers 67

security server

creating a JCL procedure (on install hosts) 7

specifying alias of security server in WLM 23

SECURITY_SERVER_ALIAS keyword in [somsec] stanza 88

SERVER general resource class

description 5

servers

adding a class to a server implementation 59

adding a class to all servers 60

adding a server 54

application server

authorizing 71

changing a server 56

configuring and running 10

creating a JCL procedure (on install hosts) 7

defining a security identity (on install hosts) 7

deleting a class from all server implementations 60

deleting a server 61

deleting one class from a server implementation 61

displaying status 51

initializing somOS::server-based application

servers 99

listing aliases 58

listing classes in the implementation repository 60

making a server available 51

multisystem considerations 92

registering in the implementation repository 53

registering with WLM 7

secure

showing all servers 58

showing one server 57

stopping 51

stopping and making a server available 51

sysplex considerations 93

service policy

activating in WLM 48

shared volumes 91, 92

simultaneous remote calls

ENCAP_POOL_SIZE keyword in [SOMD_TCPIP]

stanza 83

SMADDSTAR keyword in [somc] stanza 80

SMEMIT keyword in [somc] stanza 79

SMEMITPREFIX keyword in [somc] stanza 79

SMINCLUDE keyword in [somc] stanza 78

SMLANG keyword in [somk] stanza 84

SMNOPRINTNAME keyword in [somc] stanza 79

SMOE keyword in [somc] stanza 78

socket connection requests

capacity planning 12

capacity planning (on DSOM hosts) 106

SOM subsystem

considerations in a multisystem environment 91

considerations in a sysplex 93

creating a JCL procedure (on install hosts) 3

defining a security identity (on install hosts) 4

recycling (on DSOM hosts) 106

recycling (on install hosts) 9

specifying configuration file values 4

starting 5

SOM_TraceLevel keyword in [somras] stanza 86

SOM@CFG utility

-i option 92, 94

multisystem considerations 92

running (on DSOM hosts) 105

running (on install hosts) 9

sysplex considerations 94

SOMBASE keyword in [somk] stanza 84

sombt stanza 77

somc stanza 78

somd stanza 80

SOMD_POSSOM stanza 88

SOMD_TCPIP stanza 82

SOMDCOMMDEBUGFLAG keyword in [somed]

stanza 80

SOMDDIR keyword in [somed] stanza 80

SOMDOBJs general resource class

creating profiles 73

description 5

SOMDPORT keyword in [SOMD_TCPIP] stanza 82

SOMDPROTOCOLS keyword in [somed] stanza 81

SOMDRECWAIT keyword in [somed] stanza 81

SOMDSENDWAIT keyword in [somed] stanza 81

SOMDSVR program 99

SOMDTIMEOUT keyword in [somed] stanza 81

SOMErrorLogControl keyword in [somras]

stanza 86

SOMErrorLogDisplayMsgs keyword in [somras]

stanza 86

SOMErrorLogFile keyword in [somras] stanza 85

SOMErrorLogSize keyword in [somras] stanza 86

SOMIR keyword in [somir] stanza 83

somir stanza 83

somk stanza 84

SOMM_MVS_Secure metaclass

security considerations 67

setting up security 73

- somnm stanza** 84
- SOMNMOBJREF keyword in [somnm] stanza** 85
- somOS::server-based application servers** 99
- SOMOSSVR program** 99
- somras stanza** 85
- somsec stanza** 87
- STEPLIB concatenation** 97
- stopping a server** 51
- stopping and making a server available** 51
- stream creation for BTREE protocol**
 - BTREE_StreamCreation keyword in [SOMD_POSSOM] stanza 89
- sysplex**
 - RACF considerations 94
- sysplex, kind of sysplex required** 92
- system LXs**
 - required for &sommmvs. 2

T

- TCP/IP communications packets**
 - SOMDCOMMDEBUGFLAG keyword in [somd] stanza 80
- timeout error**
 - SOMDRECVWAIT keyword in [somd] stanza 81
 - SOMDSENDWAIT keyword in [somd] stanza 81
- trace level**
 - SOM_TraceLevel keyword in [somras] stanza 86
- trace log**
 - MVSTraceLog keyword in [somras] stanza 87
 - MVSTraceLogSize keyword in [somras] stanza 87
- trusted network** 67

U

- USER keyword in [somsec] stanza** 87

V

- validating your distributed SOMobjects configuration** 13
- validating your non-distributed SOMobjects configuration** 6
- VOLUME keyword in [sombt] stanza** 77

W

- wait time for a socket to become readable**
 - SOMDRECVWAIT keyword in [somd] stanza 81
- wait time for a socket to become writable**
 - SOMDSENDWAIT keyword in [somd] stanza 81
- WLM (workload manager)**
 - configuring SOMobjects 15
- WLM commands**
 - managing servers 51
- work classifications in WLM** 41

Communicating Your Comments to IBM

OS/390
SOMobjects
Configuration and
Administration Guide

Publication No. GC28-1851-01

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

If you are mailing an RCF from a country other than the United States, you can give the RCF to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF at the back of this book.
- If you prefer to send comments by FAX, use this number:
 - FAX: (International Access Code)+1+914+432-9405
- If you prefer to send comments electronically, use this network ID:
 - IBMLink: (United States customers only): KGNVMC(MHVRCFS)
 - IBM Mail Exchange: USIB6TC9 at IBMMAIL
 - Internet e-mail: mhvrcfs@vnet.ibm.com
 - World Wide Web: <http://www.s390.ibm.com/os390>

Make sure to include the following in your note:

- Title and publication number of this book
- Page number or topic to which your comment applies

Optionally, if you include your telephone number, we will be able to respond to your comments by phone.

Reader's Comments — We'd Like to Hear from You

**OS/390
SOMobjects
Configuration and
Administration Guide**

Publication No. GC28-1851-01

You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Today's date: _____

What is your occupation?

Newsletter number of latest Technical Newsletter (if any) concerning this publication:

How did you use this publication?

- | | | | |
|--------------------------|-------------------------------|--------------------------|------------------------|
| <input type="checkbox"/> | As an introduction | <input type="checkbox"/> | As a text (student) |
| <input type="checkbox"/> | As a reference manual | <input type="checkbox"/> | As a text (instructor) |
| <input type="checkbox"/> | For another purpose (explain) | | |

Is there anything you especially like or dislike about the organization, presentation, or writing in this manual? Helpful comments include general usefulness of the book; possible additions, deletions, and clarifications; specific errors and omissions.

Page Number:

Comment:

Name

Address

Company or Organization

Phone No.



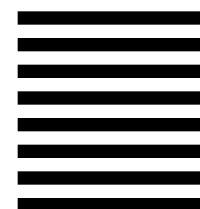
Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Department 55JA, Mail Station P384
522 South Road
Poughkeepsie NY 12601-5400



Fold and Tape

Please do not staple

Fold and Tape



File Number: S370/S390-79
Program Number: 5647-A01



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

GC28-1851-01

