ZSS Cross Memory Server

Installation Instructions

November 2018
# Table of Contents

1. Introduction
   1.1 Overview

2. Deployment
   2.1 MVS Installation
      2.1.1 Load modules
      2.1.2 PPT entry
      2.1.3 APF-authorization
      2.1.4 Started task JCL
      2.1.5 DOEPRM00 PARMLIB member
   2.2 Security requirements
      2.2.1 Required SAF classes and profiles

3. Operating ZSS Cross Memory Server
   3.1 Starting ZSS Cross Memory Server
   3.2 Modify commands
   3.3 Stopping ZSS Cross Memory Server

4. Appendix A - PARMLIB parameters

5. Appendix B – Runtime parameters
   5.1 Running multiple ZSS Cross Memory Server instances
   5.2 COLD start

© Copyright Rocket Software, Inc. or its affiliates. 2018. All rights reserved.
1. Introduction

1.1 Overview

ZSS Cross Memory Server is an authorized server application that provides privileged cross-memory services to unprivileged applications on z/OS in a secure manner.
2. Deployment

2.1 MVS Installation

2.1.1 Load modules

ZSS Cross Memory Server consists of a single load module with the name DOESRV01. This load module is supplied in the hlq.HDOE110.SDOELOAD data set as part of SMP/E install.

Note:

- This data set must be a PDSE due to language requirements.
- Where hlq is the high-level qualifier value. During the installation process, replace hlq with the data set name format that conforms to the naming standards for your installation.
- The started task must use a STEPLIB DD statement to declare the ZSS Cross Memory Server load library name. This is required so that the appropriate version of the software is loaded correctly. Do not add the hlq.HDOE110.SDOELOAD data set to the system LNKLST or LPALST concatenations.

2.1.2 PPT entry

A ZSS Cross Memory Server must run in key 4 and be non-swappable. For the server to start in this environment, a corresponding PPT entry must be added to the SCHEDXX member of the system PARMLIB. You can find an example of a PPT entry in the DOESCHED member of the hlq.HDOE110.SDOESAMP sample library.

2.1.3 APF-authorization

Due to the nature of the services ZSS Cross Memory Server provides, its load library requires APF-authorization. Before you start the server, make sure the hlq.HDOE110.SDOELOAD data set has been APF-authorized.

Example 1:

To check the APF-authorization status of hlq.HDOE110.SDOELOAD, enter:

D PROG,APF,DSNAME=hlq.HDOE110.SDOELOAD

Example 2:

To dynamically add the SMS-managed library hlq.HDOE110.SDOELOAD to the APF list, enter:

SETPROG APF,ADD,DSNAME=hlq.HDOE110.SDOELOAD,SMS
2.1.4 Started task JCL

ZSS Cross Memory Server runs as a started task. An example of the JCL can be found in the DOESRV01 member of the hlq.HDOE110.SDOESAMP sample library. You will need to copy the JCL to your system PROCLIB and adjust the STEPLIB DD according to the name hlq.HDOE110.SDOELOAD data set on your system.

The user assigned to the started task must have an OMVS segment.

See Appendix B for the runtime parameters that are accepted by the started task.

2.1.5 DOEPRM00 PARMLIB member

The ZSS Cross Memory Server started task requires a valid DOEPRMxx PARMLIB member to be found at startup. Member DOEPRM00 of the hlq.HDOE110.SDOESAMP data set contains the default configuration values. Copy this member into your system PARMLIB data set.

See Appendix A for PARMLIB parameter descriptions.

2.2 Security requirements

2.2.1 Required SAF classes and profiles

Once ZSS Cross Memory Server has been deployed and started, trusted applications can make requests for ZSS Cross Memory Server services on that system.

To protect the services from unauthorized callers, ZSS Cross Memory Server performs a sequence of SAF checks at various stages of the request.

System SAF requirements:

- The FACILITY class is active and RACLISTed

Example 1:
To see the current class settings, enter the following TSO command:

SETROPTS LIST

Example 2:
To activate the FACILITY class, enter:

SETROPTS CLASSACT(FACILITY)

Example 3:
To RACLIST the FACILITY class, enter:

SETROPTS RACLIST(FACILITY)
A valid caller of ZSS Cross Memory Server services must have the following SAF authority:

- READ access to **DOE.SERVER01.RES01** in the **FACILITY** class

Example 1:
To define the DOE.SERVER01.RES01 profiles in the FACILITY class, enter:

```
RDEFINE FACILITY DOE.SERVER01.RES01 UACC(NONE)
```

Example 2:
To grant a user READ access to the DOE.SERVER01.RES01 profile, enter:

```
PERMIT DOE.SERVER01.RES01 CLASS(FACILITY) ID(user-id) ACCESS(READ)
```

Example 3:
To refresh the FACILITY class, enter:

```
SETROPTS RACLIST(FACILITY) REFRESH
```

**Note:** ZSS Cross Memory Server will treat 'no decision' style SAF return codes as failures. If there is no covering profile for DOE.SERVER01.RES01 resource in the FACILITY class, the user will be denied.

ZSS Cross Memory Server clients, might have additional SAF security requirements. For more information, see the documentation for the specific client.
3. Operating ZSS Cross Memory Server

3.1 Starting ZSS Cross Memory Server

ZSS Cross Memory Server runs as a started task. To start a ZSS Cross Memory Server, use the standard start z/OS operator commands.

ZSS Cross Memory Server supports reusable address spaces and can be started with the REUSASID=YES keyword.

For example:

S DOESRV01,REUSASID=YES
3.2 Modify commands

ZSS Cross Memory Server supports several modify commands to display service information, enable logging, and so on. The following table lists all the commands an operator can issue.

<table>
<thead>
<tr>
<th>Command</th>
<th>Short name</th>
<th>Arguments</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPLAY</td>
<td>D</td>
<td>[OPTION_NAME]</td>
<td>Print service information</td>
<td>F DOESRV,D</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>OPTION_NAME:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• CONFIG - Print server configuration</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>information (default)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>F DOESRV,FLUSH</strong></td>
<td></td>
</tr>
<tr>
<td>LOG</td>
<td>-</td>
<td>&lt;COMP_ID&gt; &lt;LOG_LEVEL&gt;</td>
<td>Set log level</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>COMP_ID:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• CMS - Cross memory server</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• CMSPC - PC routines</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• STC - STC base</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>LOG_LEVEL:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• SEVERE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• WARNING</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• INFO</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• DEBUG</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• DEBUG2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• DEBUG3</td>
<td></td>
</tr>
</tbody>
</table>

3.3 Stopping ZSS Cross Memory Server

Use the standard z/OS operator command (P) to stop ZSS Cross Memory Server.

For example:

P DOESRV01

ZSS Cross Memory Server is designed to be a long-lived address space. There is no requirement to recycle on a regular basis.
## 4. Appendix A - PARMLIB parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOE.SECMGMT.CLASS</td>
<td>The SAF class that is used in the security services</td>
<td>DOE.SECMGMT.CLASS=XFA CILIT</td>
</tr>
</tbody>
</table>
| DOE.SECMGMT.AUTOREFRESH  | This parameter determines whether the security services automatically refresh the profiles in the DOE.SECMGMT.CLASS class after performing profile management operations. | DOE.SECMGMT.AUTOREFRESH=YES
|                          |                                                                             | DOE.SECMGMT.AUTOREFRESH=NO          |
5. Appendix B – Runtime parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Syntax</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>NAME=your_server_name</td>
<td>Server name</td>
<td>NAME=DOETESTSVR01</td>
</tr>
<tr>
<td>MEM</td>
<td>MEM=parmlib_mem_suffix</td>
<td>PARMLIB member suffix</td>
<td>MEM=02</td>
</tr>
<tr>
<td>DEBUG</td>
<td>N/A</td>
<td>Debug mode</td>
<td>PARM='DEBUG'</td>
</tr>
<tr>
<td>COLD</td>
<td>N/A</td>
<td>Discard any global data that is</td>
<td>PARM='COLD'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>associated with the server name</td>
<td></td>
</tr>
</tbody>
</table>

More detailed descriptions can be found below.

5.1 Running multiple ZSS Cross Memory Server instances

Some applications might need to use a certain version of ZSS Cross Memory Server (whereas other applications might require a newer version). To run several instances of ZSS Cross Memory Server, each unique instance must have its own name.

Specify a name using the NAME runtime parameter in the STC JCL. If you do not specify a name, the server will use the default name "DOESRV_STD".

5.2 COLD start

A ZSS Cross Memory Server instance uses common storage to make itself discoverable to the callers. If ZSS Cross Memory Server ends up in a broken state, an operator can use the COLD start runtime parameter to start the server with a clean slate.

NOTE: Contact IBM Support before using COLD start. Starting a server with the COLD start option will reset the other servers' state on the system and leak a small amount common storage.