

Adabas SOA Gateway

Installation and Operation

Version 2012-12-17

December 2012

This document applies to Adabas SOA Gateway Version 2012-12-17.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2006-2012 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, United States of America, and/or their licensors.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://documentation.softwareag.com/legal/>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://documentation.softwareag.com/legal/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices and license terms, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". This document is part of the product documentation, located at <http://documentation.softwareag.com/legal/> and/or in the root installation directory of the licensed product(s).

Document ID: APS-IMS-TBD-20121217

Table of Contents

Installation and Operation v

1 Installing the IMS/DC Client Environment in a Message Processing Region
(MPR) 1

2 User documentation for SMARTS IMS/DC interface 5

 Log Messages 6

 Online Messages 7

Installation and Operation

This documentation provides information on how to install and operate the IMS/DC Client Environment in a Message Processing Region (MPR) for the use in conjunction with Software AG's Multiple Architecture Runtime System (SMARTS).

The information on installing and operating SMARTS is structured as follows:

Installing the IMS/DC Client Environment in a Message Processing Region (MPR)

Information on how to install the IMS/DC Client Environment in a Message Processing Region (MPR)

User documentation for SMARTS IMS/DC interface

The user documentation for SMARTS IMS/DC interface

1 Installing the IMS/DC Client Environment in a Message Processing Region (MPR)

► Install the IMS/DC Client Environment in a Message Processing Region (MPR)

1. **Create a member which will hold all programs which need to be pre-loaded into the MPR.**

The member must be named DFSMPLxx where xx is the suffix characters designated by the PRLD parameter in the MPR JCL.

2. **Include list of programs which invoke SMARTS facilities.**

The pre-load program list is used to ensure that all programs that use SMARTS do not get deleted from the MPR storage during the life of their unit-of-work (transaction). Add all User-written application program names which use SoftwareAG software – e.g. Adabas SQL Server, XTS or any other required SoftwareAG optional programs.

2. **1. Create a member which will hold all programs which will need to run at MPR initialisation.**

The member must be named DFSINTxx where xx is the suffix characters designated by the PREINIT parameter in the MPR JCL.

2. **Include POSIX Initialisation Table to the list.**

The pre-initialisation program list is used to automatically initialise SMARTS POSIX Kernel at MPR start-up: Add PAINKERN to the list of programs to execute at MPR initialisation.

- 3 **Create the MPR Procedure**



Caution: All IMS applications which invoke SMARTS facilities must run in a dedicated IMS MPR.

1. **Create the JCL to invoke a MPR**
2. **Set a single dedicated Class parameter to run in this MPR – e.g. CL1=032.**
3. **Set a Pre-initialisation parameter to point to the dataset member which holds all modules to be run at MPR initialisation time – e.g. PREINIT=AQ.**
4. **Set a Preload parameter to point to the dataset member which holds all programs to be preloaded – e.g. PRLD=AP.**
5. **Include the IBM OS/390 Callable Service linkage-assist routine library Add SYS1.CSSLIB to the STEPLIB concatenation in the procedure.**

6. **Include the SMARTS load library**

Add APSvvv.MVSLD00 dataset to the STEPLIB concatenation in the procedure- where vvv is the SMARTS version.

7. **Include other SoftwareAG libraries**

Add other required SoftwareAG datasets as directed by the Installation Procedure of each product.

8. **Add the SYSPARM dataset**

Add a “//SYSPARM DD” statement for the SMARTS POSIX configuration. This dataset is required to define the runtime characteristics of your POSIX environment. For more information, see SMARTS Configuration Sources.

9. **Add a dataset for environment variables**

Add a “//CONFIG DD” statement for the dataset containing the environment variables required for your POSIX applications within IMS/DC. This DD name is specified by the ENVIRONMENT_VARIABLES parameter in SYSPARM, which defaults to “CONFIG”. For more information, see the section SMARTS POSIX Miscellaneous Parameters, ENVIRONMENT_VARIABLES.

10. **Optional SoftwareAG supplied programs**

PAINKERX – This program may be used to communicate with the SMARTS Kernel to initialize and terminate the Kernel, and to invoke Kernel commands. PAINUSNF – This performs a User sign-off of the SMARTS IMS/DC environment. This does not affect the normal running of the MPR if the associated transaction with this program is not used when a user ends an IMS session.

11. IMS Sysgen requirements

Any optional SoftwareAG programs which are to be used will have to be included in the IMS Sysgen together with the transaction names associated with these programs. The transaction names may be named to installation standards.

12. IMS/DC SMARTS interface module list

The following programs support the SMARTS IMS/DC interface:

PAIAINT
PAINENVF
PAINITPT
PAINKERN
PAINKERX*
PAINKTXT
PAINMNIT
PAINNMIT
PAINPGMT
PAINUSNF*

Where Next ?

You have now installed and configured the SMARTS IMS/DC software. All User-written IMS/DC application programs which interface with SoftwareAG software must link-edit the SMARTS Environment Dependent Initialisation module PAIAINT, and, if required, SMARTS main Environment Dependent wrapper module PAINMNIT, or SMARTS non-main Environment Dependent wrapper module PAINNMIT, into the application program load module. For more information, see the SMARTS documentation.

2 User documentation for SMARTS IMS/DC interface

| | |
|-------------------------|---|
| ■ Log Messages | 6 |
| ■ Online Messages | 7 |

APSPSX0089-nnnnnnnn SMARTS/IMS: ffffffff of token failed - yyyy zzzzzzzzz, return code n

where :

nnnnnnnnnn

system id

ffff

function - Create, Retrieve or Delete

yyyy

IMS system id

ZZZZZZZZ

token value

n

return code from MVS Token Services

| | |
|---------------|---|
| Action | Contact SoftwareAG Customer Support Desk giving the exact error message for further diagnosis and support |
|---------------|---|

APSPSX0098-nnnnnnnnn SMARTS/IMS: tt...

| | |
|--------------------|---|
| Explanation | This is a general informational message |
|--------------------|---|

where – ...

tt

informational text

Action Depending on the message the following action should be taken

6

| Message text | Reason | Action |
|--|-------------------------------|--|
| No IMS parameter list found (pppppppp) - where pppppppp is a SoftwareAG program id | IMS parameter cannot be found | Check that the IMS system is active and that the program is running in the correct MPR for SoftwareAG specific applications. Otherwise contact SoftwareAG Customer Support Desk. |

Online Messages

Issued by PAINKERX and prefixed by

APSPSX0099

- No parameters/command passed with transaction id
- No IMS parameter list found (see above for further action)
- Kernel initialised successfully
- Kernel terminated successfully
- Kernel command issued successfully
- Kernel tttttttt error - return code: cc, reason code: rr

where

tttttttt

is the request type - Initialisation, Termination or Command

cc

is the return code

rr

is the reason code



Note: refer to SMARTS documentation for information on return and reason codes

Issued by PAINUSNF

None. See SMARTS log for all messages.

Issued by PAINMNIT and prefixed by -

APSPSX0099

Program *pppppppp* terminated with return code *rr*

where

pppppppp

is the application program name

rr

is the return code (if any) returned by the application program