

Installation of SMARTS Client on CICS

Software AG recommends that you keep unmodified copies of all materials distributed or created as part of the installation process. This may assist with problem diagnosis later.

The installation procedure is described under the following headings:

- The Installation Tape
- Generic Installation Procedure
- Installing the SMARTS Server Environment
- Installing the CICS Client Environment
- Installation Verification
- Where Next ?

The configuration parameters are described in *Configuring the SMARTS Environment*.

The Installation Tape

Note:

While you are free to rename the datasets, the dataset names used in this section are used consistently throughout the product documentation to ensure clarity.

- Tape Contents
- Copying Contents of the Tape to Disk

Tape Contents

The following table lists the product datasets and what the dataset contains:

Dataset	Contains ...
APSVrs.LOAD	all load modules required by SMARTS
APSVrs.SRCE	all JCL, sample source members, and SMARTS macros.

Datasets Created during the Installation Process

Dataset	Contains ..
APSVrs.USERSRCE	source members
APSVrs.USERLOAD	load modules for all SMARTS environments
APSVrs.CICSLOAD	CICS client environment

Sample Members Copied to the APSvrs.USERSRCE Dataset

The following table lists the sample members copied to the user source dataset during the installation process. These must be modified before being used:

Member	Contains ..
RJANPARM	sample server environment parameters.
RJBNINS1	sample JCL to copy TLINOS to an authorized dataset.
RJBNPROC	sample procedure to run the SMARTS server environment.

Copying Contents of the Tape to Disk

The cartridge is delivered with a standard Software AG Tape Creation Report indicating the datasets, formats, and sizes of each dataset on the cartridge.

To create the mainframe datasets, load the datasets using either of the following:

- the IEBCOPY utility for PDS datasets
- the IEBGENER utility for sequential datasets.

Generic Installation Procedure

The following steps are required for OS/390 environments.

Step 1: Allocate and Initialize User PDS Datasets

- To create and initialize the datasets required for SMARTS, modify the sample job in member RJBNINS2 on the APSvrs.SRCE dataset to suit your installation's environment, and run it to create the appropriate datasets.

This job also copies all modifiable members from the APSvrs.SRCE dataset to the newly created APSvrs.USERSRCE dataset in order to retain all APSvrs.SRCE members as delivered.

Step 2: Specify the TCP/IP Configuration

1. Modify the CDI_DRIVER Parameter

In the SMARTS environment, the CDI_DRIVER parameter is used to specify CDI (communication driver interface) protocol definitions. If TCP/IP communications are required, a definition for protocol 'tcpip' is used.

Modify the CDI_DRIVER parameter of the sample configuration member PXANCONF .

The following section provides a sample of the CDI_DRIVER parameter specification for an IBM OE TCP/IP stack. Refer to the chapter *Configuring the SMARTS Environments* for information about the parameters that can be specified.

For an IBM OE TCP/IP Stack

```
CDI_DRIVER=('tcpip,PAAOSOCK')
```

Note:

The userid used for the job running the SMARTS environment with the IBM OE TCP/IP Stack driver configured must have an OE segment defined in the RACF (or compatible) profile.

2. Customize the SMARTS TCP/IP host name and address table

If your TCP/IP stack on OS/390 does not support host name/host address lookup (DNS), SMARTS uses a local address table that mimics the DNS functionality.

Use the sample host name parameter member PXANHOST in APSvrs.SRCE and customize to suit your needs. When customizing the local table, define:

- any host names and addresses that will be accessed from within the SMARTS server partition and
- the host where the local SMARTS server is executing.

For example, for a local host with name LOCAL and IP address 127.0.0.1 and a remote host with name REMOTE and IP address 157.189.160.95, you should specify:

```
127.0.0.1 LOCAL AF_INET 157.189.160.95 REMOTE AF_INET
```

Verify and if necessary add or modify the following parameter in the members RJBNPROC and PXANCONF to point to the PXANHOST member:

```
HOSTS_FILE=file://APSVRS/SRCE/PXANHOST
```

3. Review the Input and Output Datasets

In all environments, SMARTS requires output DD statements for STDOUT, and STDERR and an input DD statement for STDIN.

- If the output DD statements are not provided, they default to a SYSOUT dataset;
- If the input DD statement is not provided, it defaults to an empty default dataset.

Software AG recommends that you allow these DD statements to default unless the output needs to be directed in some way, or input is required on STDIN for the application in question.

Installing the SMARTS Server Environment

This section explains the additional steps required to run the SMARTS server environment.

Step 1: Authorize the SMARTS Server Environment (if necessary)

SMARTS itself does not require to be run as an authorized user, however, some SMARTS based products may have such a requirement generally or if you want to use certain features. To run SMARTS as an authorized user, proceed as follows:

1.

To achieve this, place module TLINOS in an authorized dataset. Use one of the following alternatives:

Alternative 1:

Copy the module to a library that is already authorized and use the library as the STEPLIB for SMARTS. Sample JCL to copy the module is contained in member RJBININS1.

Note:

If you do not use RJBININS1, you must ensure that whatever method you use to copy the module retains the correct AMODE/RMODE attributes and authorization code for the module.

Alternative 2:

Authorize the distributed SMARTS load library permanently by adding an entry for the dataset in member IEAAPFnn of library SYS1.PARMLIB, where "nn" is the suffix used at your site.

Note:

This requires an IPL of the system. You can avoid having to perform an IPL for the initial installation if you have a product installed that allows automatic authorization of the dataset.

2. Modify the SMARTS Server Procedure: Specify only the authorized dataset as STEPLIB, and add a COMPLIB load library concatenation containing all other load libraries, including the SMARTS load library:

```
//STEPLIB DD DISP=SHR,DSN=APF.AUTHORIZED.LIBRARY
// *
//COMPLIB DD DISP=SHR,DSN=APSVrs.USERLOAD
//APSULIB DD DISP=SHR,DSN=APSVrs.LOAD
```

Step 2: Customize the SMARTS Server Procedure

The SMARTS server environment is initialized or started by invoking a procedure or running a job with the appropriate JCL. The example JCL member RJBNPROC illustrates a typical SMARTS procedure that might be used for an OS/390 installation.

1. Add the procedure to the appropriate library

When first installing the SMARTS server environment, add the procedure RJBNPROC to the installation's system procedure library SYS1.PROCLIB or to a user-defined procedure library.

The example member contains the following JCL:

```
//RJBNPROC PROC APSPARM=RJANPARM,
//          OPARM=,
//          REG=32M
// *****
```

```

/*
/* SMARTS Server Environment Startup Procedure for OS/390
/*
/******
//IEFPROC EXEC PGM=TLINOS,
//      PARM='&OPARM;',
//      REGION=&REG,TIME=1440,DPRTY=(14,14)
/*
//STEPLIB DD DISP=SHR,DSN=APSVrs.USERLOAD
//APSULIB DD DISP=SHR,DSN=APSVrs.LOAD
/*
//SYSPARM DD DISP=SHR,DSN=APSVrs.USERSRCE(&APSPARM;)
//SYSPRINT DD SYSOUT=X

```

2. Prepare the procedure

During the installation process, either

- alter the supplied RJBNPCOC procedure to suit your requirements and copy the resulting procedure to an installation procedure library.

3. Invoke the procedure

Invoke the SMARTS procedure either from the operator's console using

- an OS/390 START command

```
S RJBNPCOC,...
```

- or an OS/390 batch job

```

//SMARTS JOB .....
//IEFPROC EXEC RJBNPCOC,...

```

In either case, you need to understand the DD statement functions and the use of available start-up options to implement the features of the SMARTS server environment.

The following sections use the above sample JCL as a basis for defining the SMARTS server environment initialization procedure for OS/390.

4. Check required and optional DD statements

The required and optional DD statements in the above procedure are described in more detail below:

STEPLIB	Required . Identifies the authorized load library in which the SMARTS server environment OS/390 start-up module TLINOS resides. No other SMARTS server environment modules need to be available in this library. This dataset is only referenced once during initialization and therefore its placement is not an issue.
COMPLIB	Required . Identifies the PDS library concatenation that effectively becomes the STEPLIB for the duration of the run. This means that all modules loaded during the execution of the SMARTS server environment are loaded from this concatenation.
SYSMDUMP	Optional . Identifies where the OS/390 control program should write a formatted dump if an ABEND occurs or the SMARTS server environment requests such a dump. If not specified, no support can be provided for any problem as no diagnostic information is available. Software AG recommends that you specify SYSMDUMP instead of the SYSABEND or SYSUDUMP DD statements, as more information is available for diagnosis if a problem should occur. Note as well that a SYSMDUMP usually completes in less time. Estimates for the size of the dataset specified by this DD statement must be made according to the IBM documentation.
SYSPRINT	Optional . Identifies where statistics are printed at termination or using the STATS operator command. If not specified, this DD is allocated using dynamic allocation as a sysout dataset.
SYS Parm	Required . Identifies the file or library member in which the desired SMARTS server environment system parameters are to be found. If the SMARTS server environment is to be periodically stopped and started in order to test various start-up options, the symbolic parameter &RTSPARM can be used to identify the member containing the desired options. However, you can use the PARM option at start-up time as a short-term test of a specific option. The available system parameters are described in Configuring the SMARTS Environments.

5. Check the symbolic parameters

You can modify the SMARTS procedure, including the format and use of the symbolic parameters. However, the symbolic parameters indicated below are generally sufficient to meet the needs of most installations:

&OPARM	Specifies a character string that is passed to the SMARTS server environment control program using the PARM sysparm (see the chapter Configuring the SMARTS Environments for use of this feature).
&RTSPARM	Specifies the member name in the library identified by the DD name SYSPARM that contains the control statements specifying the start-up parameters (sysparms). You can create multiple members to allow tailoring of SMARTS server environment initialization to meet the specific conditions defined by the control statements.

Step 3: Add SMARTS Server Configuration Statements

- Add the appropriate SMARTS server configuration parameters prior to starting the SMARTS server environment.

In particular, select a dataset name pattern for DUMPDSN= that will allow the user in effect for the SMARTS address space to allocate names.

The configuration parameters are described in *Configuring the SMARTS Environment*.

For parameters relevant to your application, refer also to the documentation for the software that runs on SMARTS.

Installing the CICS Client Environment

Step 1: Modify the CICS Procedure

1. Add the LOAD datasets to the DFHRPL concatenation

Add the APSvrs.LD00 dataset to the DFHRPL concatenation in the procedure.

2. 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*.

3. Add a dataset for environment variables

Add a `//CONFIG DD` statement for the dataset containing the environment variables required by your POSIX applications within CICS.

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.

4. Provide a DFHZNEP node error program

If the installation already has a DFHZNEP node error program in use, modify it to invoke the SMNE transaction under the conditions detailed in the model assembler program PACNZNEP, supplied in the APSvrs.SRCE Tape file. .

If the installation does not have a DFHZNEP node error program in use, use the supplied model program PACNZNEP to create one.

Step 2: Define Transactions to CICS

- Four standard transactions are required:
 - TDSP, which is used internally by SMARTS;
 - SMGO, which is used to initialize the POSIX server;
 - SMEX, which may be used to terminate the POSIX server;
 - SMNE, which is invoked by the CICS Node Error Program to provide SMARTS cleanup functionality.

The standard transactions must be defined to run the following Assembler programs:

- TDSP to run PACNSTRT. The name TDSP is mandatory as it is used internally by SMARTS. Define this Transaction with TASKDATALOC(BELOW) and TASKDATAKEY(USER)..
- SMGO to run PACNKERN. The name SMGO is mandatory as it is used internally by SMARTS. Define this Transaction with TASKDATALOC(BELOW) and TASKDATAKEY(CICS)..
- SMEX to run PACNKERX. SMEX is a suggested transaction name. Define this Transaction with TASKDATALOC(BELOW) and TASKDATAKEY(CICS).
- o SMNE to run PACNNEP. The name SMNE is mandatory, as it is used internally by SMARTS. Define this Transaction with TASKDATALOC(BELOW) and TASKDATAKEY(CICS)..

Step 3: Define Transactions to CICS

1. Autoinstall should also be activated. Autoinstall can be activated by setting the SIT parm :

```
PGAIPGM=ACTIVE
```

2. The following Assembler language programs must be defined to CICS.

- PACNABEX with EXECKEY(CICS)
- PACNKERN with EXECKEY(CICS)
- PACNKERX with EXECKEY(CICS)
- PACNNEP with EXECKEY(CICS)
- PACNPCEX with EXECKEY(USER)
- PACNSMGO with EXECKEY(CICS)
- PACNSTRT with EXECKEY(CICS)

The following Assembler Programs must be defined to CICS as follows:

- RAANPARM, define with "RELOAD=YES", and EXECKEY(USER)

Step 4: Define Programs to the Program List Tables

1. POSIX Initialization Table

The program list table PLTPI is used to automatically initialize SMARTS POSIX at CICS startup:

In the DFHPLTxx for the PLTPI, insert PACNSMGO as a second phase PLT program.

2. POSIX Shutdown Table

The program list table PLTSD is used to automatically terminate SMARTS POSIX at CICS shutdown:

In the DFHPLTxx for the PLTSD, insert PACNKERX as a first phase PLT program.

Installation Verification

1) Initialization Messages of SMARTS

A successful initialization of SMARTS can be verified in the CICS protocol output and looks similar to this messages:

```
+APSPSX0015-* POSIX V271 Build 030212 Patch level=2 Initialization in prog
+APSPSX0004-* Module 'L$CPRSU' Loaded
+APSPSX0050-SysName CDI FILE protocol initialized
+APSPSX0066-SysName Trace level = 1
+APSPSX0068-SysName No System Tracing enabled
+APSPSX0069-SysName No functions are being traced
+APSPSX0036-SysName Global environment variables processed successfully
+APSPSX0026-SysName Sockets Initialization successful
+APSPSX0050-SysName CDI TCPIP protocol initialized
+APSPSX0064-SysName Trace DataSpace Initialised, ESIZE=0:BSIZE=0:NBLKS=0
+APSPSX0065-SysName Log DataSpace Initialised, ESIZE=0:BSIZE=0:NBLKS=0
+APSPSX0008-SysName SMARTS SERVER V271 System initialized, nucleus size 56
+PACNKERN - POSIX INTERFACE INITIALISED.
```

2) Installation Verification

Verification of communication between CICS using Natural and EntireX Communicator can be performed by logging on the Natural library SYSETB and proceeding the Natural Tutorial .

Example – CICS:

```
N411 RCA=(BROKER),RCALIAS=(BROKER,EXAAPSC)
```

3) CICS Shutdown verification

A successful shutdown of CICS using the "CEMT P,SHUT" command will present PACNKERX - POSIX INTERFACE TERMINATED on the terminal that issued the shutdown command. This verifies that control has been given to the SMARTS termination routine.

Where Next ?

You have now installed the SMARTS software. You can continue now with the installation of the application that is to run on SMARTS.

Note that the configuration procedure of the application that runs on SMARTS may instruct you to modify some of the configuration parameters of SMARTS.