Configuration of the SMARTS Server Environment

Start-up parameters are available to customize the execution of the SMARTS server environment. The start-up options, whether specified as PARM parameters or entered as statements read from SYSPARM (z/OS) or SYSIPT (VSE), are specified as keyword parameters (so-called "sysparms") and must be entered according to established keyword coding conventions. See Sysparm Format.

The sysparms are interpreted and processed by the PARM-processor module of the SMARTS server environment when the server environment is initialized.

Note:
For z/OS and VSE systems, SMARTS server sysparms may be overridden during the initialization of the environment without updating the member in the partitioned dataset. For more information about specifying or overriding sysparm data, see Installation on z/OS and Installation on VSE.

For a standard z/OS installation, Software AG recommends that you define the size of the installation by setting the region size and one or both of the parameters

```
WORKLOAD-AVERAGE
WORKLOAD-MAXIMUM
```

and omit the configuration of the following parameters to SMARTS:

ADABAS-BP
BUFFERPOOL
ROLL-BUFFERPOOL
SAVEPOOL
SAVEPOOL-ANY
TASK-GROUP
TIBTAB
THREAD-GROUP

If one or more of these sysparms are set, these settings will be used.

The following parameters may be specified in a SMARTS server environment only; otherwise, the following warning message will be issued:

```
Unknown keyword = xxxxxxxx
```

where *xxxxxxxx* is the keyword that was not recognized.

SMARTS Server Configuration Parameters

ADABAS-BP

Sysparm	Use	Possible Values	Default
ADABAS-BP	Used to define the ADABAS buffer pool.	see text	The number of subpools to allocate will be decided internally. The size of the elements depends on the definitions made to create ADALCO .

This buffer pool is used for ADABAS interface work areas, which are acquired outside of the thread but in the key of the thread. This parameter enables users to determine the key(s) for which buffer subpools are built and the number of buffers in each subpool.

The format for the value is as follows:

```
ADABAS-BP=((no,key),(no,key) . . . (no,key))
```

- where

no	is the number of elements to allocate in the buffer subpool for this key. This must be greater than 1 and less than or equal to 8192.
key	is the storage protect key in which the buffer subpool will be allocated. This may be any number between 1 and 15. For z/OS, FACOM, and Hitachi systems, only keys 8 to 15 should be specified here.

By default, a subpool is built for keys 8 to 15. 8192 bytes are allocated for each subpool and the number of areas that can exist in each subpool is dependent on the size of the various ADALNK areas required.

Notes:

1. If an error is encountered in an ADABAS-BP system parameter, the whole line of code is ignored. Therefore, if there is no following ADABAS-BP specification in the system parameters, the defaults are in effect.
2. A subsequent specification of the ADABAS-BP system parameter totally overwrites a previous ADABAS-BP specification. Therefore, if the second specification is incorrect, the defaults again apply even if the first ADABAS-BP specification is correct.

Example:

```
ADABAS-BP=((20,9),(50,12),(100,8))
```

An ADABAS buffer pool is built with three subpools:

- the first is built in key 9 and has 20 elements;

- the second is built in key C(12) and has 50 elements; and

- the third is built in key 8 and has 100 elements.

ADACALLS

Sysparm	Use	Possible Values	Default
ADACALLS	The maximum number of incomplete ADABAS calls from an application before the SMARTS/ ADABAS interface rolls out the application.	see text	10

Note:
This parameter is ignored if ADAROLL=NO is specified.

The format for the value is as follows:

```
ADACALLS={ n | (dbid,n) }
```

- where

n	is an integer between 1 and 32767.
dbid	is an ADABAS database ID. If 'dbid' is specified, ADACALLS applies only to calls directed to the specified database.

ADADBID

Sysparm	Use	Possible Values	Default	Required
ADADBID	The default database ID for ADABAS.	1-255	none	no

The value specified for ADADBID is used if the application program does not supply a specific database ID in the ADABAS control block. Refer to the *ADABAS Operations Manual* for a description of the use of the database ID.

ADALIMIT

Sysparm	Use	Possible Values	Default
ADALIMIT	The maximum number of ADABAS calls from an online transaction before the program is cancelled.	see text	4096

Note:
This parameter is ignored for attached programs.

The format for the value is as follows:

```
ADALIMIT={ n | (dbid,n) }
```

- where

n	is an integer between 0 and 32767. If ADALIMIT=0 is specified, this parameter is ignored (no limit).
dbid	is an ADABAS database ID. If 'dbid' is specified, ADALIMIT applies only to calls directed to the specified database.

Specifies the maximum number of ADABAS calls from an online transaction without an intervening terminal I/O. Programs that exceed this limit are cancelled with error message ADA0003.

ADAROLL

Sysparm	Use	Possible Values	Default
ADAROLL	The number of seconds the SMARTS server environment will wait for ADABAS calls before it rolls out the program making the call.	see text	see text

The format for the value is as follows:

```
ADAROLL={ n | (dbid,n) | ALWAYS | (dbid,ALWAYS) | NO | (dbid,NO) }
```

- where

n	is an integer representing the number of seconds that the SMARTS server environment will wait.
dbid	is an ADABAS database ID. If 'dbid' is specified, ADAROLL applies only to calls directed to the specified database.
ALWAYS	indicates that the program is always eligible for rollout.
NO	indicates that the program is never eligible for rollout.

By default, the SMARTS server environment dynamically calculates the optimum value for each database based on the statistics for the database. The starting value is ALWAYS; i.e., at the first ADABAS call, the program is always eligible for rollout. Then ADAROLL is calculated based on the average response time (A) using the following rule:

A < 0.05 sec	ADAROLL=0.1
0.05 sec < A < 0.5 sec	ADAROLL=2*A
A > 0.5 sec	ADAROLL=ALWAYS

Software AG recommends that you allow this parameter to default.

ADASVC

Sysparm	Use	Possible Values	Default
ADASVC	The decimal SVC number to be used when communicating with ADABAS.	see text	13

The format for the value is as follows:

```
ADASVC={ n | (dbid,n) }
```

- where

n	is an integer 201 to 255 for z/OS, and 1 to 110 for VSE.
dbid	is an ADABAS database ID. If 'dbid' is specified, ADASVC applies only to calls directed to this database.

By default (ADASVC=13), the interface to ADABAS version 5 or above is disabled. Programs issuing a call to ADABAS version 5 or above are terminated with ABEND code U0004.

APPLYMOD

Sysparm	Use	Possible Values	Default	Required
APPLYMOD	Include or remove a system-wide modification in/from the SMARTS session.	91 or 92	none	no

This sysparm must be used for FACOM systems. Software AG does not recommend using it in other operating system environments.

The format for the value is as follows:

```
APPLYMOD={ n | (n,NO) }
```

- where

n	is the applymod number, in this case either 91 or 92.
n,NO	indicates the removal of the 'n' applymod.

Possible Applymods

When an error other than an application program error occurs in SMARTS, a dump is normally scheduled by the SMARTS recovery processing. The dump is written to the SYSUDUMP, SYSABEND, or SYSMDUMP DD statement using normal OS rules.

- **Applymod 73: force operating system dump**
  Specify Applymod=73 only at the request of your support representative to force an operating system dump prior to recovery after an abend.

  This will cause a dump to be taken according to the installation dump options set for SMARTS. Please note, this Applymod is only required to produce additional diagnostics in an error situation.

  The installation could suffer severe performance problems, and large numbers of dumps written, if this Applymod is set for any length of time and therefore it should only be set at the request of your support representative.

- **Applymod 91: Use the OS SNAP function to write a dump**
  Applymod 92: Use the IEATDUMP or SDUMP function to write a dump
  By default, applymod 92 is in effect, ensuring that unformatted dumps are written to dynamically allocated datasets according to the dataset name pattern defined by parameter DUMPDSN= .

  This is the dump format expected by Software AG support when you send in a dump for problem analysis. Also, this is by far the fastest method of writing a dump. If you prefer to get abend dumps according to your SYSMDUMP, SYSUDUMP, or SYSABEND definitions, specify APPLYMOD=(92,NO) in order to turn off applymod 92.

  If you prefer to produce dumps using the SNAP function, specify APPLYMOD=91 *in addition*.

  Note:
  SMARTS may write dumps for certain non-abend error situations also. These dumps cannot be written to SYSMDUMP, SYSUDUMP, or SYSABEND, therefore, they are always written using either IEATDUMP / SDUMP or SNAP.

  Note to FACOM users: Software AG recommends that you also use the default (SDUMP). Software AG support may ask you to format the dump using a batch job before sending it to Software AG.

BUFFERPOOL

Sysparm	Use	Possible Values	Default
BUFFERPOOL	Defines the parameter for building the general buffer pool.	see text	see text

The format for the value is as follows:

```
BUFFERPOOL={ esize,enum [,expnum] [,loc] }
```

- where

esize	Required. Determines the size of each individual element in this buffer subpool. The value is rounded up to the next multiple of 64.
enum	Required. Determines the number of elements of the specified 'esize' that will initially be built in the buffer subpool to be defined.
expnum	Optional. Determines the number of elements by which the buffer subpool is expanded if the primary 'enum' is not sufficient. The default 'expnum' value is "enum/4" with a minimum of "1". The 'expnum' value is affected by the amount of space required for preemptive expansion of the subpool. As not all requests can expand a subpool when it becomes empty, SMARTS requires preemptive expansion of the general buffer pool. The space required for preemptive expansion is calculated internally. When the space available in the subpool reaches that specified for preemptive expansion, the subpool is expanded by one quarter of the number of subpool elements, or 10, whichever is lower. The 'expnum' value must be equal to or greater than the figure used for preemptive subpool expansion. If the specified value is lower, it is forced to this figure.

loc	Optional. Determines where the buffer subpool elements are to be allocated. Valid values are BELOW, ANY, and DS:
	
	● **BELOW**
	    the default; storage is to be allocated below the 16-megabyte line.
	
	● **ANY**
	    available only on 31-bit-capable systems; storage can be allocated anywhere within the primary address space and is to be allocated above the 16-megabyte line under normal circumstances.
	
	● **DS**
	    available only on ESA-capable systems; storage can be allocated within a data space.

For each correctly specified BUFFERPOOL parameter, a subpool is built in the general buffer pool from which all non-specific buffer pool requests are satisfied.

If the BUFFERPOOL parameter is not specified, or is specified one or more times and all are incorrect in the sense that they are unusable, SMARTS builds a default bufferpool with standard sizes and numbers of elements based on the size of your installation.

When at least one BUFFERPOOL parameter is accepted as valid, the default is not invoked. This means that the parameters are not merged.

DUMPDSN

z/OS only.

Sysparm	Use	Possible Values	Default	Required
DUMPDSN	A data set name pattern to be used for the dump data set when SMARTS writes a dump using the z/OS IEATDUMP service	Any valid data set name, use of system symbols is permitted. See IBM documentation of the macro IEATDUMP, parameter DSNAD=	If DUMPDSN= is not specified, SMARTS uses SDUMP (if running APF-authorized) or SNAP instead of IEATDUMP, otherwise no dump will be taken.	no

Software AG recommends that you do specify this parameter for non-APF authorized installations in order to avoid SNAP dumps.

For authorized installations, this is not necessary, because SDUMP can be used. SDUMPs are written to SYS1.DUMPxx (or substitute).

Note:
The data set name must be one that the userID in effect for the SMARTS address space is permitted to allocate.

Example for fixed dump data set name:

```
DUMPDSN=DUMP.DATASET.NAME
```

Example using symbols:

```
DUMPDSN=DUMP.&jobname..D&YYMMDD..T&HHMMSS.
```

EOJ-VER

Sysparm	Use	Possible Values	Default	Required
EOJ-VER	The indicated character string must be entered as part of the EOJ operator command when SMARTS terminates.	1 to 8-character string	none	no

GLOBAL-MAXENQS

Sysparm	Use	Possible Values	Default
GLOBAL-MAXENQS	The maximum number of ENQs or LOCKs that can be outstanding from user programs in the SMARTS region or partition.	100-32767	1024

INIT-PGM

Sysparm	Use	Possible Values	Default
INIT-PGM	Specifies the name(s) of programs to be loaded by SMARTS at the end of initialization.	see text	none

The format for the value is as follows:

```
INIT-PGM={ name | (name1, name2, ..., namen) }
```

The programs named are called from the nucleus during startup in the order they are specified, executed in the SMARTS address space in SMARTS's key, and deleted after execution. If a program ABENDs, SMARTS initialization optionally continues.

The programs are internal to Software AG applications that run on SMARTS and are supported by this parameter for legacy reasons.

Otherwise, it is preferable to use the SERVER statement to obtain control during startup, termination, and for operator commands, if required.

INSTALLATION

Sysparm	Use	Possible Values	Default
INSTALLATION	A 1 to 8-position character string used as an installation identification name. The name may not contain a comma.	character string	********

MAXENQS

Sysparm	Use	Possible Values	Default
MAXENQS	The maximum number of z/OS ENQs or VSE LOCKs that may be outstanding for any one application program.	1-256	15

Each outstanding ENQ/LOCK resource held occupies 24 bytes plus the length of RNAME in the general buffer pool while the resource is held (whether it is held as SHR or EXCLUSIVE).

MAXTASKS

Sysparm	Use	Possible Values	Default
MAXTASKS	The maximum number of tasks to be used within a given SMARTS run.	n	254 (z/OS, Facom, Hitachi) 27 (VSE)

- where 'n' is the maximum number of tasks that will be allocated within task groups:

- For z/OS and MSP (FACOM), the number must be greater than zero and less than or equal to 254. This is a nominal maximum of 256 less the 2 SMARTS system tasks.

- For VSE, the number must be greater than zero and less than or equal to 27; that is, VSE maximum tasks = 32 less 5 SMARTS system tasks.

This parameter should be allowed to default unless there is a valid reason for restricting the number of tasks to be attached. The only mechanisms for attaching tasks are through the start-up parameters or through the TASKS operator command.

MESSAGE-ID

Sysparm	Use	Possible Values	Default
MESSAGE-ID	Value to be used as the system ID in the SMARTS message prefix.	x (see below) | INSTALLATION	patch character

SMARTS messages have a prefix with the format

```
pppgggnnnnx
```

- where

ppp	product ID (APS)
ggg	message group ID
nnnn	message number
x	system ID

By default, the patch character is used as the system ID (see the PATCHAR sysparm).

Specify MESSAGE-ID=INSTALLATION to use the installation ID instead of the patch character as the system ID.

PATCHAR

Sysparm	Use	Possible Values	Default
PATCHAR	Except for an asterisk (*), a character that uniquely identifies the running SMARTS server environment within the system.	<char> | *	*

- where '<char>' is any valid printable character except an asterisk (*).

If another SMARTS server environment with the same patch character is active, SMARTS is terminated during initialization.

The default patch character '*' (asterisk) allows multiple SMARTS server environments with this patch character to be active at the same time.

This character is important in two areas:

1. Every message sent to the console has the patch character of the issuing SMARTS server environment following the message-identifier; for example, RTSABS0006-2. Before the sysparms are processed, the default patch character is shown in all messages.

2. Data can be added to the profile system as being specific to a certain system. When the data is read, the system searches for data relating to the patch character of the running system before taking the global information. In this way, you can customize your sessions differently in different SMARTS server environments using the same SMARTS system dataset.

PROGRAMISD

Sysparm	Use	Possible Values	Default
PROGRAMISD	The number of in-storage directory (ISD) slots to be reserved for SMARTS online programs.	n	100

- where 'n' is an integer from 1 to 16 digits in length. The minimum value is 10.

Each program ISD entry occupies 128 bytes of page-fixed storage containing the disk address of an online program that has been or is executing. For a given ISD, the entries are dynamically altered to reflect the most current program usage based upon frequency of use.

RESIDENTPAGE

Sysparm	Use	Possible Values	Default
RESIDENTPAGE	The name of a program to be loaded and made resident when SMARTS is initialized.	program-name	none

This parameter is relevant only for Com-plete. In all other environments, all modules are assumed to be reentrant, and are loaded into the address space automatically at first reference.

The program must be fully reentrant. If it is not marked reentrant, a warning message is issued on the operator's console at SMARTS initialization time.

The program must reside in the COMPLIB chain (z/OS) or the LIBDEF search chain (VSE) of the SMARTS initialization procedure.

ROLL-BUFFERPOOL

Sysparm	Use	Possible Values	Default
ROLL-BUFFERPOOL	The size of the fixed roll buffer pool	(Esize,Eno,Expno,Loc)	not allocated

The values have the same meaning as for the BUFFERPOOL parameter, except for Loc. The following values are valid for the ROLL-BUFFERPOOL Loc:

BELOW	allocate the roll buffer pool below the 16MB line only
ANY	allocate the roll buffer pool either below or above the 16MB line
DS	(the default) allocate the roll buffer pool in a data space

SAVEPOOL

Sysparm	Use	Possible Values	Default
SAVEPOOL	The number of "savepool" entries to be allocated below the 16MB line.	n>=100	calculated by SMARTS depending on the configuration

SAVEPOOL is a critical parameter as these areas are used as base level save areas and can therefore not be expanded. If they are filled, SMARTS terminates abnormally.

SAVEPOOL-ANY

Sysparm	Use	Possible Values	Default
SAVEPOOL-ANY	The number of "savepool" entries to be allocated above the 16MB line.	n>=100	calculated by SMARTS depending on the configuration

It is important to carefully review the value specified for SAVEPOOL-ANY based on the usage of the system. When these areas run out, the system can continue to run using savepool entries allocated below the line; however, this wastes a valuable resource.

SECSYS

Sysparm	Use	Possible Values	Default
SECSYS	An alternate security subsystem to validate user IDs and passwords during logon.	NO | RACF | ACF2 | TOPSECRET | COMSEC,R|A|T	NO

The specified subsystem is interrogated to determine dataset access authority during utility processing. This parameter applies to z/OS.

SECSYS-APPL

Sysparm	Use	Possible Values	Default
SECSYS-APPL	The application name to be used for uniquely identifying this SMARTS nucleus to the external security system (see SECSYS).	name	SAG#RTS

SERVER

Sysparm	Use	Possible Values	Default
SERVER	Information that identifies a server to SMARTS.	server-information	none

- where the server information has the format

```
  (serv-id , init-mod , p1 , p2 .... pn)
```

serv-id	is the ID for this server (1-8 chars)
init-mod	is the name of the initialization/termination routine
p1...pn	are parameters to be passed to the initialization routines

Specifying the SERVER parameter causes SMARTS to build a server directory entry (SDE) for the specified server and pass control to the initialization routine specified to initialize the server.

STARTUPPGM

Sysparm	Use	Possible Values	Default
STARTUPPGM	Specifies the name(s) of one or more SMARTS application programs to be invoked at the end of initialization.	see text	none

The format for the value is as follows:

```
  STARTUPPGM={ name | (name1, name2, ..., namen) }
```

These programs are scheduled in the order in which they are specified to execute in SMARTS server threads once the system has initialized. These application programs execute as attached tasks under SMARTS's user ID and can use all SMARTS functionality.

Sufficient batch or free TIBs must be available in SMARTS's TIBTAB to accommodate the number of programs specified.

Note:
Each process running in a SMARTS server environment has a control block called TIB associated with it. The TIB contains identifying information such as a one- to five-digit terminal identification number (TID) and a one- to eight-character terminal information block name (TIBNAME). Either the TID or TIBNAME may be used to specify a single process.

TASK-GROUP

Sysparm	Use	Possible Values	Default
TASK-GROUP	A group comprising one or more tasks, available when SMARTS is started.	(grp,num,priority,maxq)	(DEFAULT,num)

- where

grp	Required. The name of the task group being defined. The default task group is DEFAULT.
num	Required. The number of tasks to be allocated in the task group. This value must be greater than 1 and less than 254 (z/OS) or 27 (VSE). The default number of tasks is calculated dynamically based on the size of the installation.
priority	the priority to be assigned to the operating system task, which is attached for z/OS and MSP (FACOM) systems only. This parameter is accepted under VSE, but has no meaning. Valid values are 0-255; the default is 248. '255' is the priority at which the task-dependent service processor task is running. Without the ADABAS high performance environment (HPE), this is '250'. While '255' is accepted, the task will in fact only be given a priority of '250'.
maxq	The maximum number of TIBs (default 16) expected on this task group's work queue at the same time. Under normal circumstances, the default should be adequate. When there are problems and it is not, a secondary Last In First Out (LIFO) queue is used so that no work is lost. The normal queue is First In First Out (FIFO), which ensures that work is done in the order in which it is received. This is why the LIFO queue is only used as a secondary backup.

Important:
For SMARTS, only the TASK-GROUP DEFAULT is available. Software AG strongly recommends that you use the default definition. If other products running on SMARTS require changes to the defaults or allow the definition of their own TASK-GROUPs, that will be indicated in the relevant documentation.

Notes:

1. A maximum of 8 task groups may be defined.
2. Task-group names are converted to uppercase prior to being processed; therefore, a parameter entered in lowercase is treated as, and appears in, uppercase letters.
3. If more than one specification appears for a task group, the last valid specification is used.
4. The task group DEFAULT must always exist in the system. If it is not explicitly defined by the installation, the task group is built by the system with the default values.
5. Note that the total number of tasks to be attached must not exceed the MAXTASKS specification. This is not checked until the task groups are being built; however, exceeding the value leads to task-group allocation errors as against parameter errors.

Examples:

```
TASK-GROUP=(DEFAULT,4)
```

The DEFAULT task group is allocated with four attached tasks, the default priority, and the default maximum queue size specification.

```
TASK-GROUP=(DEFAULT,4,200)
TASK-GROUP=(TASK-GRP,4,150)
```

The DEFAULT task group is allocated with four attached tasks with a priority of 200 and the default maximum queue size specification. A second group called TASK-GRP is also allocated with three attached tasks, a priority of 150, and the default maximum queue size specification.

THREAD-GROUP

Sysparm	Use	Possible Values	Default
THREAD-GROUP	A thread group containing one or more thread subgroups and threads, to be available when SMARTS is started.	see below	see below

The format for the value is

```
 (grp,(sub,size,num,cpu,real,key),...,(sub,size,num,cpu,real,key))
```

- where

grp	Required. The name of the thread group being defined.
sub	The name of the subgroup being defined. If a subgroup name is specified more than once for the same group, the last valid specification is used when parameter processing has been completed.
size	Required. The amount of storage in kilobytes to be allocated for each thread below the line. A valid value is between 8 kilobytes and 1 megabyte.
num	The number of threads to be allocated in the thread subgroup. The value must be greater than 1 and less than 4096. Generally, this subparameter is required. It can be omitted for one (and only one) thread subgroup in the address space; in this case, the number of threads to be allocated for the subgroup is calculated dynamically by SMARTS based on the size of the installation.
cpu	The CPU time in seconds (default 0.00) that a user program can use in the thread subgroup for one SMARTS transaction. This value may be entered as an integer or to a level of hundredths of seconds using the 'n.nn' format. If a 0 is provided as the CPUTIME for a thread subgroup, no CPU limit is placed on programs running in the associated threads.
real	The wait time in seconds (default 0.00) for the thread subgroup, after which a message is issued to the console if the user program has not given up control of its thread. This value may be entered as an integer or to a level of hundredths of seconds using the 'n.nn' format. If 0 is specified, elapsed time is not checked for the thread subgroup.
key	The key (default M) in which the threads within the subgroups are allocated: ● **M** The thread keys are a mixture of user keys excluding the key in which SMARTS is running. ● **N** No storage protection is implemented and all threads run in the same key as SMARTS. **Note:** The user may also specify a value in the range 1 to 15 inclusive to allocate a thread to that key explicitly.

The default value is

```
THREAD-GROUP=(DEFAULT,($DEFAULT,8,num))
```

- where "num" is calculated dynamically based on the size of the installation.

Important:
For SMARTS, only the THREAD-GROUP DEFAULT is available. Software AG strongly recommends that you use the default definition. If other products running on SMARTS require changes to the defaults or allow the definition of their own THREAD-GROUPs, that will be indicated in the relevant documentation.

Notes:

1. A maximum of 8 thread groups may be defined.
2. A maximum of 8 subgroups can be allocated per thread group. The subgroups may be defined on one line or on different lines. When a second THREAD-GROUP statement is used, the same group name must be specified to relate the subgroup entries.
3. Thread group and subgroup names are converted to uppercase prior to being processed; therefore, a parameter entered in lowercase is treated as, and appears in, uppercase letters.
4. If more than one specification appears for a thread subgroup of a thread group, the last valid specification is used.
5. The amount of storage specified on the THSIZEABOVE sysparm is allocated above the line for each thread defined as a result of the THREAD-GROUP sysparm.
6. The thread group DEFAULT must always exist in the system. If it is not explicitly defined by the installation, the thread group is built by the system with the default values. If it is defined, the system ensures that a thread subgroup with a thread size at least as large as that required by DEFAULT is allocated. If not, the system allocates an additional subgroup for the group. If too many subgroups have been defined, the last one defined is overwritten to allow for the default specification.
7. The keyword data is processed from left to right. If more than one thread subgroup is defined on one line and the line contains an error, even if an error message is issued for the line, any subgroups processed up to the error are still accepted. That is to say, if the first subgroup is correct and the second is not, an error message is issued but the first thread subgroup is defined while the second and subsequent specifications in the same statement are ignored.

Examples

```
THREAD-GROUP=(DEFAULT,(SMALUTIL,80,3),(BIGUTIL,300,2,5,9,15))
```

This allocates the DEFAULT thread group with two subgroups:

- the first subgroup called SMALUTIL contains three threads with 84K below the line and takes the defaults for CPUTIME, REALTIME, and the protectkey to be allocated to the thread.

- the second subgroup called BIGUTIL contains two threads with 304K below the line, has a maximum CPUTIME of 5 CPU seconds, a REALTIME value of 9 seconds, and each thread has a storage protectkey of 15.

The following sets of sysparms defines exactly the same thread subgroups:

```
THREAD-GROUP=(DEFAULT,(SMALUTIL,80,3),(BIGUTIL,300,2,5,9,15))
THREAD-GROUP=(DEFAULT,(SMALUTIL,40,8),(BIGUTIL,300,2,5,9,15))
THREAD-GROUP=(DEFAULT,(SMALUTIL,80,3))
```

The following sets of sysparms defines exactly the same thread subgroups in two thread groups, one called DEFAULT and the other called EXTRAGRP:

```
THREAD-GROUP=(DEFAULT,(SMALUTIL,80,3))
THREAD-GROUP=(EXTRAGRP,(BIGUTIL,300,2,5,9,15))
THREAD-GROUP=(EXTRAGRP,(SMALUTIL,80,3))
THREAD-GROUP=(DEFAULT,(BIGUTIL,300,2,5,9,15))
```

THSIZEABOVE

Sysparm	Use	Possible Values	Default
THSIZEABOVE	The amount of storage above the 16 MB line, in multiples of 1024 bytes, to be allocated to each thread.	n	1024

TIBTAB

Sysparm	Use	Possible Values	Default
TIBTAB	The location and size of the TIB table to be built when SMARTS is initialized.	DYNnnnnn | ANYnnnnn	ANYnnnnn

- where

DYN	is the table to be built below the 16MB line
ANY	is the table to be built above the 16MB line
nnnnn	is the number of TIBs. The maximum is 32767. For the default value, "nnnnn" is calculated dynamically based on the size of the installation.

Note:
Each process running in a SMARTS server environment has a control block called TIB associated with it. The TIB contains identifying information such as a one- to five-digit terminal identification number (TID) and a one- to eight-character terminal information block name (TIBNAME). Either the TID or TIBNAME may be used to specify a single process.

TRACECLASS

Sysparm	Use	Possible Values	Default
TRACECLASS	The class of trace event to be included in (or excluded from) the SMARTS trace table.	class | (class,OFF)	QTIB

- where 'class' is one of the following valid trace classes:

GENERIC	used for support purposes
QTIB	TIB queue management
OP	application program requests
FIXBPOOL	fixed-length buffer pool operations
ROLL	roll-processing events
RESOURCE	resource manager get/free
DISPATCH	dispatcher events

The option '(class,OFF)' indicates exclusion of the specified class.

TRACEOPTION

Sysparm	Use	Possible Values	Default
TRACEOPTION	The specified trace option is active for this execution of SMARTS.	option	no options active

The valid trace options are as follows:

ABEND	The trace continues to run during SMARTS abnormal termination. (Normally the trace stops recording at the first indication of termination.) Use only when required by Software AG support personnel.
EXTENDED	Trace processing uses the extended form of the trace record. Use only when requested by support personnel to find specific information as it decreases the number of trace records that can be held in a trace buffer.

TRACETABLE

Sysparm	Use	Possible Values	Default
TRACETABLE	The size of the SMARTS trace table, which is used to trace events occurring within the SMARTS system.	n | nK	8K

The TRACETABLE sysparm can be a valuable tool for problem resolution.

The minimum size of the trace table is 8K.

TRACETABLE=0 indicates that no tracing is performed.

WORKLOAD-AVERAGE
WORKLOAD-MAXIMUM

The WORKLOAD-AVERAGE parameter specifies a normal workload value, and the WORKLOAD-MAXIMUM parameter specifies a maximum workload value. SMARTS uses these values togther with the region sizes above and below the 1bMB line to configure itself.

These parameters are not required, but tuning them may improve performance.

Sysparm	Use	Possible Values	Default
WORKLOAD-AVERAGE	The average number of parallel processes expected to run in SMARTS	1-32767	WORKLOAD-MAXIMUM divided by 4
WORKLOAD-MAXIMUM	The maximum number of parallel processes expected to run in SMARTS	1-32767	50 if WORKLOAD-AVERAGE is not specified, otherwise WORKLOAD-AVERAGE times 4

Example:

```
WORKLOAD-AVERAGE=50
WORKLOAD-MAXIMUM=400
```