



---

	Version 2010-05-31 18:34:06 (MET)	Messages and Codes
--	--------------------------------------	--------------------



This document applies to RisAris SOA Gateway 2010-05-31 18:34:06 (MET) and to all subsequent releases.

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

© Copyright RisAris 2010.

All rights reserved.

The name RisAris Limited and/or all RisAris Limited product names are either trademarks or registered trademarks of RisAris Limited. Other company and product names mentioned herein may be trademarks of their respective owners.

## Table of Contents

1 SMARTS Messages and Codes .....	1
2 SMARTS Messages - Format and General Introduction .....	3
Message Format .....	4
Message Documentation .....	4
3 SMARTS Abend Codes .....	7
SMARTS API ABEND Codes (Prefix U) .....	8
SMARTS Server Environment ABEND Codes .....	10
4 SMARTS API Messages .....	11
5 SMARTS Server Environment Messages .....	49
ABE – ABEND Processing .....	50
ABS – Service Routine Processing .....	53
ADA – ADABAS Interface .....	57
BPM – Buffer Pool Management .....	59
DIS – Dispatching Mechanism Messages .....	63
INI – Initialization : Main Processing .....	71
INP – Initialization : Parameter Processing .....	75
LOD – Program Management Services .....	76
OPC – Operator Communication .....	78
OUS – ";User" Operator Command Processor .....	96
RES – Reentrant Program Support .....	97
ROL – Rollout / Rollin Processing .....	100
RSM – Resource Management .....	103
STG – Storage Initialization .....	104
SVR – Server Processing .....	106
TIB – Terminal Initialization .....	108
TMR – Timing Services .....	109
ZDM – COMDMP Dump Dataset Processing(VSE Only) .....	114
ZLA – Security and Accounting .....	117
ZOS – OS Initialization .....	117
ZTR – Trace Utilities .....	119
ZTS – Thread Storage .....	119
A Allocated Error Numbers (errno) .....	121
B SMARTS Server Environment Internal Macro Description .....	125
ATTACH Function .....	126
CANCEL Function .....	126
CMTIMR Function .....	127
ABEXIT Function .....	127
C Request Status Codes .....	129
Return Codes .....	130
Feedback Codes .....	130



# 1 SMARTS Messages and Codes

---

This documentation describes the SMARTS messages and codes issued by the SMARTS system. Some messages and codes appear in all SMARTS environments; other messages and codes appear only in the SMARTS server environment.

The documentation also contains errno mapping tables for the various errors returned by program functions and the various CDI modules used by SMARTS.

This information is organized as follows:

• <b>Introduction</b>	Explains message format and structure .
• <b>SMARTS Abend Codes</b>	Messages with prefix U, ABEND
• <b>SMARTS API Messages</b>	Messages with prefix APSPSX
• <b>SMARTS Server Environment Messages</b>	Messages with prefix ABE, ABS, ADA, BPM, DIS, INI, INP, LOD, OPC, OUS, RES, ROL, RSM, STG, SVR, TIB, TMR, ZDM, ZLA, ZOS, ZTR, ZTS
• <b>Allocated Error Numbers</b>	Each errno value that may be returned to a PAANHLL request, its equivalent C macro name and a short description of what the errno means.
• <b>SMARTS Server Environment Internal Macro Description</b>	Return codes from SMARTS internal functions.
• <b>Request Status Codes</b>	Return and feedback codes indicating processing results of the request.

---

## 2 SMARTS Messages - Format and General Introduction

---

- Message Format ..... 4
- Message Documentation ..... 4

This documentation contains messages and codes issues by SMARTS.

## Message Format

---

All SMARTS messages have the following format:

```
APSsssnnnn - message-id
```

-where

sss	identifies the subsystem issuing the message.
nnnn	is a sequential message number identifying the message within the subsystem.
message-id	identifies the SMARTS address space issuing the message, which is determined from the SMARTS server environment configuration parameter MESSAGE-ID. The value may be a single character in braces (e.g. (X)) or it may contain the installation ID as specified in the SMARTS server environment configuration parameter INSTALLATION. Refer to SMARTS Installation and Operations for more information.

## Message Documentation

---

Messages are documented with the message identifier (excluding the constant `APS' which is always present) in the heading followed by the message text. The message text has the format

```
This is a test &1 with three (&2) replacement parms `&3'
```

When displayed or written to the console, the message contains the text as displayed here; however, the placeholders identified by the `&n' construct are replaced by data relevant to the message. If the data to be displayed is X, y and AB, the message appears as

```
This is a test X with three (y) replacement parms `AB'
```

Each message has the following subsections:

- Description

Describes why the message was issued.

- Placeholders

Describes the contents of the placeholders; i.e., the `&n' values in the message. If the message contains no placeholder, this section contains the expression `Not applicable'.

■ Action

Describes the actions to be taken when the message is issued. A number of actions may be listed in this section, if appropriate. If no action is required, this section contains the expression `Not applicable`.

■ Additional References

Describes additional sources of information that may help to explain the message. If no additional references are available, this section contains the expression `Not applicable`.

---

# 3 SMARTS Abend Codes

---

- SMARTS API ABEND Codes (Prefix U) ..... 8
- SMARTS Server Environment ABEND Codes ..... 10

Under normal circumstances, SMARTS attempts to terminate a user program with a descriptive message when an error occurs; however, this is not always possible and an ABEND must be issued to terminate the application in these cases.

This document covers the following topics:

## SMARTS API ABEND Codes (Prefix U)

---

### U0000 – Insufficient Storage

**Explanation** Insufficient storage is available to obtain a primary control block within the SMARTS environment. A problem occurred at a point where it was not possible to issue a SMARTS error message. A request for storage from the environment failed.

**Action** Make more storage available to the program running in the environment so that the request can be satisfied.

### U0001 - Logic Error in SMARTS Nucleus

**Explanation** An unexpected condition occurred within the nucleus for which a message could not be issued. The ABEND may indicate storage overwrites or earlier errors in the SMARTS environment.

**Action** If the problem cannot be determined, report the ABEND to your Software AG technical support representative.

### U0002 - Storage Corrupted

**Explanation** While freeing storage within the SMARTS nucleus, a corruption failure was detected. When storage is allocated, SMARTS puts a storage accounting prefix at the start of the storage and an identical suffix at the end of the storage. If these do not match when the storage is freed, this ABEND occurs. A mismatch indicates that either

- |  |
|--|
| <ul style="list-style-type: none"> <li>the user that allocated the storage requested ";n" bytes but used more than ";n" bytes, overwriting the storage accounting area at the end; or</li> </ul> |
| <ul style="list-style-type: none"> <li>a user of storage before the storage area being freed overwrote the prefix of the storage area.</li> </ul>  |

**Action** Register 7 will point to the prefix accounting area of the storage. The length of the storage is found at register 7 + 4 (if it hasn't been corrupted), while the storage area itself returned to the user can be found at register 7 + 8.

**U0003 - SMARTS Not Initialized**

**Explanation** An attempt was made to issue a SMARTS API request in an environment where SMARTS was not initialized. This ABEND occurs in multiuser environments (such as Com-plete or CICS) if SMARTS has not been initialized prior to running a SMARTS-based program. It may occur in a single-user environment, such as batch, if an attempt to dynamically initialize SMARTS fails.

**U0004 - Invalid SMARTS Main Control Block**

**Explanation** The SMARTS interface stubs found an anchor pointer in the environment where the program is running; however, this anchor pointer points to a control block that is not the main SMARTS control block. This may occur if the incorrect environment-dependent stubs have been linked with the application.

**U0005 - No SMARTS Context Found**

**Explanation** An attempt to locate the existing SMARTS context or create a new SMARTS context failed.

**Action** Refer to previously issued messages to determine why this failed.

**U0006 - ABEND Due to Message Option**

**Explanation** A SMARTS message was issued for which an internal ABEND option was set.

**Action** Refer to the message and correct the problem causing the message to be issued.

**U0007 - Invalid Local Storage Stack Area Detected**

**Explanation** During internal entry or exit processing, an invalid internal stack was detected. This indicates storage corruption in the SMARTS environment or an internal error.

**Action** If there is no evidence of storage corruption, report the error to your Software AG technical support representative.

**U0009 - User Program Stack Overflow**

**Explanation** An attempt to allocate storage from the user program stack failed as there was insufficient storage available to allocate the space requested.

**Action** Increase the stack length available to the user program and retry.

**U0010 - Recursive ABEND During Termination**

**Explanation** An ABEND occurred, which the SMARTS clean-up handlers have trapped; however, during SMARTS clean-up processing, another ABEND occurred. SMARTS has therefore terminated its clean-up cycle to avoid a recursive ABEND situation. It is likely that SMARTS resources will not be cleaned up correctly if this occurs.

**Action** Determine the cause for both the primary and secondary ABEND and resolve the issues that caused them. This could occur if there were errors in a pthread\_cleanup() or atexit() registered user clean-up function.

**U0011 - Signal Generated ABEND**

**Explanation** A signal occurred for the pthread or process for which no signal handler has been registered. The default action for the signal is to terminate the process, which results in this ABEND.

## SMARTS Server Environment ABEND Codes

---

The SMARTS server environment may terminate abnormally with any of the following user ABEND codes:

**ABEND 4, 7, 9, 10, 11, 13, 14, 15**

**Explanation** These ABEND codes indicate unexpected errors in the processing of the SMARTS server environment savearea pool.

**Action** Save all related documentation including dumps and logs. Contact your Software AG technical support representative.

**ABEND 51**

**Explanation** An unexpected error occurred during processing of an error message.

**Action** Save all related documentation including dumps and logs. Contact your Software AG technical support representative.

# 4 SMARTS API Messages

---

**PSX0000    Insufficient storage (\$1 bytes) for \$2**

**Description** An attempt by the SMARTS nucleus to acquire storage failed due to insufficient space either in a user program thread or in the SMARTS address space itself.

**Placeholders**

\$1	The number of bytes the system tried to acquire. The number may be suffixed with `K` to denote kilobytes or `M` to denote megabytes. If this value is `0`, SMARTS was not able to determine how much storage could not be acquired. This could occur when a request to an underlying system fails due to a storage shortage but does not indicate how much is required.
\$2	A character string indicating what the storage was for and, if applicable, a four digit code in braces indicating which control block this storage was for.

**Action**

- |   |
|---|
| <ul style="list-style-type: none"><li>• If the storage is thread-related, use the ULIB utility to increase the catalog size of the root application that suffered the error. The root application is the first program to receive control when a program is started under SMARTS and is the basis upon which the storage is allocated. Note that the thread sizes in general may need to be increased in the SMARTS region depending on how much thread space the application requires.</li><li>• If the storage is outside of the thread, there is a shortage of storage in the SMARTS region itself. Where possible, SMARTS will expand its storage areas so it is likely that such an expansion request failed thus resulting in this message. Check for other errors related to any attempted expansion for more details.</li></ul> |
|---|

## References

- The SMARTS Installation and Operations Manual discusses storage allocations and how they relate to the SMARTS components.
- The following table identifies the storage areas by name, where it may be located and what it is used for:

Storage Name	Location	Description
atexit() registration table	Above	Allocated per process to hold any exit routines registered using the atexit() function.
File subsystem I/O buffer	Below or Above	One of these control blocks is acquired from the SMARTS buffer pool per open file, socket, or pipe. It is required as an input/output buffer to access the underlying subsystem entity. If required for a file, it is allocated below the line. For any other use, it is allocated above the line.
File table storage	Above	This storage is allocated per SMARTS process to hold the file descriptors for each open file in the system.
Kernal storage stack	Above	Allocated by the environment-dependent kernel interface module to drive the SMARTS independent kernel interface.
Working storage	Above	A generic term for required temporary storage.
SMARTS server environment ABEND data area (ABDA)	Above	Allocated per SMARTS pthread in the SMARTS server environment to handle recovery processing for pthread termination (normal or abnormal) in the server environment.
SMARTS server application storage (APPS)	Above	Allocated per SMARTS-based server started within the SMARTS server environment.
Argument buffer (ARGB)	Above	One allocated per invoked process to hold arguments passed to main(), environment variables passed to invoked process (addressable through environ), and pointers.
CDI parameter block (CDIB)	Above	One allocated per open file descriptor in a process to maintain state of the file descriptor and pass information to/from CDI protocol drivers.
CDI PH main anchor block (CDIM)	Above	One allocated per defined CDI protocol handler (driver). Used to hold CDI protocol-related information.
CDI PH process anchor block (CDIP)	Above	One allocated per defined CDI protocol handler per process to hold process-related protocol information.
CDI PH thread anchor block (CDIT)	Above	One allocated per defined CDI protocol handler per thread to hold thread-related protocol information.

Storage Name	Location	Description
Console interface block (CNIB)	Above	Allocated when an fopen() is issued to the console CDI driver.
Console interface command block (CNIC)	Above	Allocated in order to deliver a command to the console CDI driver interface.
Console interface main block (CNIM)	Above	Allocated to centrally control the console interface CDI driver processing.
Environment variable storage (ENVV)	Above	One of these areas is allocated per environment variable. If the environment variable is a global environment variable, it is allocated from the SMARTS buffer pool. If the environment variable is being locally defined using the `putenv` function, it is allocated from the application program thread.
File subsystem control block (FSCB)	Below or Above	One of these control blocks is acquired from the SMARTS buffer pool per open file, socket, or pipe. If required for a file, it is allocated below the line. For any other use, it is allocated above the line.
Internal security control block (ISCB)	Above	Allocated per SMARTS process to hold user-related information.
Language interface process table (LIPT)	Above	Allocated per process to handle languages calling other languages.
Language interface thread table (LITT)	Above	Allocated per pthread to enable language cleanup for each language used within the process.
Master control block (PMCB)	Below	The main control block allocated by SMARTS. Acquired from the underlying SMARTS server environment work storage buffer pool.
Mutex control block (PMXB)	Above	Allocated internally to control access to internal resources using the standard pthread mutex processing logic.
Process control block (PPCB)	Above	The SMARTS process control block, which is allocated once in the program thread the first time a SMARTS function request is issued.
Signal raise control block (PSGR)	Above	Allocated when a signal has to be raised either from the system internally or as a result of a user program request.
Thread control block (PTHB)	Above	Allocated per SMARTS pthread to represent the pthread.
Thread-specific data area (PTSD)	Above	Allocated by the user program per SMARTS pthread to hold any thread-specific dataset.
Thread cancel clean-up table (PTTH)	Above	Allocated per SMARTS pthread in all environments to hold each entry pushed on to the stack using the pthread_cleanup_push() function.

Storage Name	Location	Description
Sockets main control block (SOCB)	Above	Allocated from the SMARTS buffer pool by the subsystem specific sockets initialization module.
System-resident control block (SRCB)	Above	One of these control blocks is allocated from the SMARTS buffer pool per active SMARTS process. It is associated with the PPCB and is used to hold per process storage areas, which must reside outside the application program thread area.
Sockets task API storage (TAIS)	Above	One of these control blocks is allocated for each SMARTS process that uses the sockets interface. It is only allocated if the IBM TCP/IP stack is in use.

**PSX0001 Logic error in program \$1 at offset \$2**

**Description** During execution, the SMARTS nucleus encountered an unexpected situation that is likely to lead to other errors or unexpected behavior.

**Placeholders**

\$1	Name of the program where the logic error was encountered.
\$2	Offset within the program named in \$1 where the logic error was encountered.

**Action** Report this message to your technical support representative.

**References**

Not applicable.

**PSX0002 Function code '\$1' not supported**

**Description** The SMARTS nucleus API was entered with a code for which it had no entry in its internal tables. This is indicative of a mismatch between the SMARTS stubs being used and the nucleus.

**Placeholders**

\$1	Code (in decimal) with which the nucleus was entered.
-----	---

**Action** Report this message to your technical support representative.

**References**

Not applicable.

**PSX0003 Function '\$1' code '\$2' not implemented**

**Description** The function named by placeholder \$1 has been requested by the application but is not currently implemented within the SMARTS nucleus.

**Placeholders**

\$1	Name of the function which the application requested.
\$2	Internal SMARTS code associated with the function.

**Action** Not applicable.

**References**

Refer to the SMARTS SDK Programmer's Reference Manual to determine which function calls are supported by the level of SMARTS you are running.

**PSX0004 Module '\$1' Loaded**

**Description** The module identified by the \$1 placeholder was loaded by the nucleus. This message is issued in the following cases:

•	When more than one version of a module exists, it indicates which version of the module was loaded.
•	For exits that may not normally be part of the nucleus, it indicates when an exit has been loaded and is active in the system.

**Placeholders**

\$1	Name of the module which has been loaded.
-----	---

**Action** This is an informational message, no action is necessary.

**References**

Not applicable.

**PSX0005 Module '\$1' not found**

**Description** The module identified by the \$1 placeholder cannot be found. A request to the operating system to load a module fails. Modules to be loaded by SMARTS must be available either in the COMPLIB DD concatenation or system LNKLST for OS/390 and MSP systems, while it must be in a library identified in the search path for VSE.

If this message is issued during the initialization process, initialization fails if the module is required for the correct operation of SMARTS. Otherwise, initialization continues. If this message is issued during termination processing, termination continues; however, depending on the function of the module, the termination process may not complete successfully.

**Placeholders**

\$1	Name of the module that could not be found.
-----	---

**Action** If the module should be available during initialization and/or termination processing, determine why it cannot be found.

**References**

Not applicable.

**PSX0006 Error loading module `1' RC=2 Reason=x3**

**Description** The module identified by the \$1 placeholder could not be loaded due to an error during LOAD processing. A request to the operating system to load a module fails for some reason other than the fact that module cannot be found.

If this message is issued during the initialization process, initialization fails if the module is required for the correct operation of SMARTS. Otherwise, initialization continues. If this message is issued during termination processing, termination continues; however, depending on the function of the module, the termination process may not complete successfully.

**Placeholders**

\$1	Name of the module for which the LOAD request failed.
\$2	Return code from the operating system LOAD request.
\$3	Reason code from the operating system LOAD request.

**Action** Determine from the return and reason codes why the LOAD request failed and correct the error.

**References**

•	MVS/ESA Programming: Assembler Services Reference
•	VSE/ESA Macro Reference

**PSX0007 Module 1 returned return code 2**

**Description** A number of modules are called internally during the SMARTS initialization/termination process. These modules generally issue a zero (0) return code to indicate that they have completed successfully. A module was called and its return code was not 0. The module responsible has issued a message itself to indicate where the problem lies.

When this occurs during the initialization process, if the return code is less than 8, initialization processing continues. If the return code is 8 or greater, initialization processing terminates.

When this occurs during the termination process, processing continues; however, if the return code is 8 or greater, there may be additional failures later in the termination process.

**Placeholders**

\$1	Name of the module that returned the return code.
\$2	Return code returned by the module identified by \$1.

**Action** Refer to preceding messages in the log to determine why the return code was returned. Correct the situation to prevent the message.

**References**

Not applicable.

**PSX0008** **POSIX \$1 system initialized, nucleus size \$2 bytes**

**Description** The POSIX subsystem has been successfully initialized.

**Placeholders**

\$1	Name of the POSIX system as specified on the SMARTS server environment SERVER configuration parameter.
\$2	Size (in bytes) of the SMARTS nucleus. Since most SMARTS modules are loaded above the 16MB line, this storage is allocated above the 16MB line.

**Action** No action is necessary as this is an informational message.

**References**

Not applicable.

**PSX0009** **SMARTS \$1 initialization failed**

**Description** The SMARTS initialization failed. A preceding message indicates why it failed.

**Placeholders**

\$1	Name of the POSIX system as specified on the SERVER configuration parameter.
-----	--

**Action** Correct the reason for the initialization failure. Restart SMARTS.

**References**

Not applicable.

**PSX0010    POSIX \$1 system already active****Description** An attempt to start the POSIX subsystem failed because it was already active.**Placeholders**

\$1	Name of the POSIX system as specified on the SERVER configuration parameter.
-----	--

**Action** Not applicable.**References**

Not applicable.

**PSX0012    POSIX \$1 system terminated****Description** The POSIX subsystem has terminated.**Placeholders**

\$1	Name of the POSIX system as specified on the SERVER configuration parameter.
-----	--

**Action** Not applicable.**References**

Not applicable.

**PSX0013    Unrecognized command '\$1'****Description** An operator command to SMARTS was rejected as it was not a valid SMARTS operator command.**Placeholders**

\$1	Operator command string entered by the operator. SMARTS did not recognize it as a valid SMARTS operator command.
-----	--

**Action** Select a valid SMARTS operator command.**References**

The SMARTS Installation and Operations Manual .

**PSX0014 Unrecognized or invalid parameter \$1**

**Description** SMARTS was initialized with \$1 as a parameter but SMARTS did not recognize the parameter. Initialization continues, ignoring the invalid parameter.

**Placeholders**

\$1	String passed to SMARTS as a parameter. This is specified either as part of the SERVER configuration parameter or as the SERVER operator command used to start SMARTS.
-----	--

**Action** Select a valid SMARTS parameter.

**References**

The SMARTS Installation and Operations Manual .

**PSX0015 POSIX \$1 Build \$2 Patch level=\$3 Initialization in progress**

**Description** The POSIX subsystem started its initialization processing.

**Placeholders**

\$1	Name of the POSIX system as specified on the SERVER configuration parameter.
-----	--

**Action** Not applicable.

**References**

Not applicable.

**PSX0016 Function '\$1' failed return code \$2 (x'\$3')**

**Description** The function \$1 was issued by the application; however, an error during the processing of the function in the SMARTS nucleus caused the \$2 return code to be generated. A preceding message indicates why the return code was returned.

**Placeholders**

\$1	Name of the function issued by the application program.
\$2	Internal return code (decimal) issued by the processing routine.
\$3	Internal return code (hexadecimal) issued by the processing routine.

**Action** Determine from the preceding messages what caused the problem and correct these symptoms.

**References**

Not applicable.

**PSX0019 Error opening '\$1' errno=\$2**

**Description** SMARTS attempted to open the file identified by \$1, however, the request failed due to the errno \$2. The result of this message depends on when the failure occurred and what the file was to be used for.

**Placeholders**

\$1	Name of the file that could not be opened in the format used by the 'fopen' function as described in the SMARTS SDK Programmer's Reference Manual.
\$2	Error number returned by the open request. These are described in the SMARTS SDK Programmer's Guide.

**Action** Determine why the file could not be opened using the errno returned and correct the problem. Generally this should occur when either the file does not exist or there is insufficient storage to open the file.

**References**

•	SMARTS SDK Programmer's Reference Manual
•	SMARTS SDK Programmer's Guide

**PSX0020 POSIX process initialization failed**

**Description** The first time a POSIX function is issued from a program, SMARTS initializes the environment for the process where the application program is running. This message indicates that this processing failed and therefore the function requested by the program could not be completed.

**Placeholders**

Not applicable

**Action** Determine from any preceding messages why the initialization process failed and correct the problem.

**References**

Not applicable

**PSX0022 Socket \$1 request error PID=\$2 rc=\$3 reason=\$4 diag=\$5**

**Description** A sockets subsystem call failed.

**Placeholders**

\$1	Name of the TCP/IP request that failed.
\$2	Process ID on which the failure occurred.
\$3	TCP/IP interface return code returned by the request.
\$4	TCP/IP interface reason code returned by the request.
\$5	TCP/IP interface diagnostic code returned by the request.

**Action** Determine from the information returned why the error occurred. There are any number of reasons why errors will occur on a sockets call, many due to the peer to an ongoing SOCKETS conversation terminating or dropping the conversation.

**PSX0023 BATCH ENVIRONMENT \$1 ERROR, INFO=\$2/\$3/\$4**

**Description** An error occurred with an operating system macro in the SMARTS batch interface.

**Placeholders**

\$1	Name of the operating system macro that failed.
\$2	Return code from the operation (normally register 15).
\$3	Feedback code from the operation (normally register 0).
\$4	Reserved for future use; currently, the same as \$2.

**Action** Determine the cause of the problem based on the operating system macro return and feedback codes; take action to correct the problem. If the action to take is not obvious, report the error to your Software AG technical support representative.

**References**

Refer to the manuals specific to your operating system that describe Assembler interface macros and their use.

**PSX0024 ESTAE Processing failed for \$1 - RC=\$2**

**Description** An OS/390 ESTAE macro was issued but failed with a \$2 return code. Processing may continue; however, if abends occur, no recovery is in place.

**Placeholders**

\$1	Function for which the ESTAE was issued.
\$2	Return code from the OS/390 operating system ESTAE macro.

**Action** Based on the return codes, determine why the ESTAE failed and correct the error.

**References**

MVS/ESA Assembler Programmers Macro Reference Manual

**PSX0025 Sockets \$1 failed**

**Description** The subsystem-specific SOCKETS initialization or termination did not complete successfully. If initialization processing fails, SMARTS fails to initialize. If termination processing fails, termination processing continues; however, subsequent errors may occur. In particular, the entire SMARTS address space should be brought down before attempting to restart the SMARTS as there are likely to be sockets problems if the address space is not cycled.

**Placeholders**

\$1	Indicates whether 'INITIALIZATION' or 'TERMINATION' processing failed.
-----	--

**Action** Refer to preceding TCP/IP subsystem-specific error messages to determine why the failure occurred and correct the problem.

**References**

Not applicable.

**PSX0026 Sockets \$1 successful**

**Description** Sockets initialization or termination processing completed successfully.

**Placeholders**

\$1	Indicates whether 'INITIALIZATION' or 'TERMINATION' processing was successful.
-----	--

**Action** Not applicable.

**References**

Not applicable.

**PSX0029 IBM TCP/IP \$1 failure AS=\$2 errno=\$3**

**Description** An error occurred for a request made by the IBM TCP/IP subsystem module.

**Placeholders**

\$1	Name of the IBM TCP/IP application programming interface (API) request for which the error occurred.
\$2	Name of the IBM TCP/IP address space with which SMARTS is interfacing. This is specified using the SMARTS ADDRSPCE configuration parameter.
\$3	Error number returned by the IBM TCP/IP interface

**Action** Determine from the information returned why the error occurred. There are any number of reasons why errors will occur on a sockets call, many due to the peer to an ongoing SOCKETS conversation terminating or dropping the conversation.

**References**

SMARTS Installation and Operations Manual IBM TCP/IP VvRr for MVS Programmers  
Reference

**PSX0030 POSIX \$1 is quiescing**

**Description** The POSIX subsystem is quiescing. In this state, application programs already started may continue to completion; however, no new requests to start the SMARTS programs are allowed.

**Placeholders**

\$1	Name of the POSIX system as specified on the SERVER configuration parameter.
-----	--

**Action** Not applicable.

**References**

Not applicable.

**PSX0031 Waiting for \$1 user(s) to terminate**

**Description** The POSIX subsystem cannot terminate correctly until all application programs that have requested SMARTS functions have terminated. This message indicates the number of users of SMARTS functions upon which the server is waiting.

**Placeholders**

\$1	The number of users of SMARTS applications still active who are using or have used SMARTS functions.
-----	--

**Action** Wait until all users have terminated and reissue the request to QUIESCE or TERMINATE the POSIX subsystem. The server may be forced; however, this is not recommended due to the subsequent problems it can cause.

**References**

Not applicable.

**PSX0033 Waiting on \$1 dependent service(s)**

**Description** When subsystems initialize and have a dependency on SMARTS, they register this dependency. For example, the HTTP server may have one or more active servers in a given system. If these dependent services have not been terminated prior to SMARTS termination, SMARTS issues this message and will not terminate.

**Placeholders**

\$1	Number of servers in the system that have registered their dependency on SMARTS but have not removed this dependency by terminating.
-----	--

**Action** Terminate any dependent services prior to requesting the termination of SMARTS. SMARTS may be forced; however, this is not recommended due to the subsequent problems it may cause and the immediate problems it will cause for dependent services.

**References**

Not applicable.

**PSX0034 I/O Error (errno=\$1) reading \$2**

**Description** While reading the file identified by \$2, an error was returned by the I/O subsystem interface module. Processing of the file will be terminated.

**Placeholders**

\$1	Error number returned by the I/O subsystem interface module as documented in the SMARTS SDK Programmer's Guide.
\$2	Name of the file upon which the I/O error occurred. This is in the format used by the 'fopen' function as described in the SMARTS SDK Programmer's Reference Manual.

**Action** Determine from the returned information what caused the error and correct the problem.

**References**

SMARTS SDK Programmer's Reference Manual. SMARTS SDK Programmer's Guide.

**PSX0035 Environment variable '\$1' invalid data starting '\$2'**

**Description** While processing the file containing the SMARTS global environment variables, a variable containing invalid data was encountered. This generally occurs when a hexadecimal value is specified for a global environment variable.

**Placeholders**

\$1	Name of the global environment variable with the invalid data specified.
\$2	The first characters of the value specification as found in the file.

**Action** Correct the value specification for the variable.

**References**

SMARTS Installation and Operations Manual.

**PSX0036 Global environment variables processed successfully**

**Description** The global environment variable file as specified by the GENVNAME SMARTS configuration parameter was processed successfully and the global environment variables specified in the file have been successfully defined.

**Placeholders**

Not applicable.

**Action** Not applicable.

**References**

SMARTS Installation and Operations Manual.

**PSX0037 DYNALLOC FAILED (CODE=\$1), DSN=\$2**

**Description** The SMARTS attempt to allocate dataset (\$2) failed.

**Placeholders**

\$1	Error and information codes returned from the OS DYNALLOC macro.
\$2	Name of the dataset that SMARTS tried to allocate.

**Action** Determine the cause of the problem based on the OS DYNALLOC macro error and information codes; take action to correct the problem. If the action to take is not obvious, report the error to your Software AG technical support representative.

**References**

Refer to the manuals specific to your operating system that describe Assembler interface macros and their use.

**PSX0038 TISP/ISP \$1 failure TID=\$2 RC=\$3 Reason=\$4**

**Description** TISP/ISP macro call failed.

**Placeholders**

\$1	Name of the TISP/ISP request that failed.
\$2	Process ID on which the error occurred.
\$3	TISP/ISP interface return code returned by the request.
\$4	TISP/ISP interface reason code returned by the request.

**Action** Use the information returned to determine why the error occurred. Errors on a TISP API call often occur when the peer to an ongoing TISP API conversation terminates or drops the conversation.

**References**

SMARTS Installation and Operations Manual. The Fujitsu manual TISP Handbook .

**PSX0040 R=\$1 T=\$2 U=\$3 S=\$4 RC=\$5 errno=\$6**

**Description** This message is used when sockets tracing is active and the IBM TCP/IP subsystem module is in use.

**Placeholders**

\$1	Name of the IBM TCP/IP sockets request.
\$2	Address of the TCB on which the request was issued in hexadecimal format.
\$3	Number of sockets open for this process ID.
\$4	Number of the IBM TCP/IP socket for which the request was issued.
\$5	Return code from the IBM TCP/IP sockets request.
\$6	Error number returned from the IBM TCP/IP sockets request.

**Action** Not applicable.

**References**

IBM TCP/IP VvRr for MVS Programmers Reference.

**PSX0041 Context locate/initialization failed rc=\$1**

**Description** An attempt to build a new SMARTS context failed. Each program running under SMARTS, including the SMARTS main task itself, requires a context. When the context build fails, no further processing can occur.

**Placeholders**

'RC' is the return code:

8	Error during context initialization.
---	--------------------------------------

**Action** The reason why the build for the context failed is indicated in a preceding message. Take action to correct the problem based on that information.

**References**

Not applicable.

**PSX0042 Independent Kernel \$1 Request failed rc=\$2**

**Description** An attempt by the environment-specific module to make a request of the independent kernel interface failed. This can happen during SMARTS kernel initialization or termination, or while processing an operator command. If the error occurs during initialization, processing cannot continue.

**Placeholders**

\$1	The request that failed:	
	initialize	initialization request
	terminate	termination request
	command	command processing
\$2	Return code:	
	8	general error
	12	catastrophic error
	16	logic error within SMARTS

**Action** Take action based on a previous message that indicates why the request failed. Report return codes 12 and 16 to your Software AG technical support representative.

**References**

Not applicable.

**PSX0043 Resource \$1 not authorized for user \$2, reason \$3**

**Description** Access to a resource (for example, a file) has been denied by the active security subsystem.

**Placeholders**

\$1	Name of the requested resource.
\$2	Name of user requesting access to resource.
\$3	An implementation-specific reason code.

**Action** Use the implementation-specific reason code to determine the cause of the denied access.

**References**

Not applicable.

**PSX0044 Invalid length supplied to '\$1' function. Program will be aborted**

**Description** A number of functions are supported using inline coding generated by the C compiler that generated the C code. Normally, any length below 16 megabytes is dealt with inline; however, if a length greater than this is encountered, it is considered an error and this message is issued.

**Placeholders**

\$1	The name of the function that received an invalid length.
-----	---

**Action** Trace back to the C code and determine where the length supplied was invalid; determine why the length was invalid.

**References**

Not applicable.

**PSX0045 pthread \$1 \$2 failure rc=\$3**

**Description** An error occurred during the initialization or termination of a pthread.

**Placeholders**

\$1	Initialization / termination processing
\$2	PAANTINI / PMANENVF end-of-job (EOJ) indicator
\$3	Return code

**Action** Normally, a previous error indicates the reason for the error. Take action based on this information. If no reason is provided, report the problem to your Software AG technical support representative.

**References**

Not applicable.

**PSX0046 Pthread id \$1 \$2**

**Description** A pthread ABENDED or was cancelled.

**Placeholders**

\$1	The pthread ID.
\$2	Cancelled / ABENDED.

**Action** A pthread\_cancel() request from the application indicates that the application has taken the decision, or an abend in a pthread indicates an error in the application code running on the pthread. Otherwise, an attempt was made to cancel all outstanding pthreads when a SMARTS instance was terminated.

**References**

Not applicable.

**PSX0047 DXR Server environment \$1 error, info=\$2/\$3/\$4**

**Description** An error occurred with an operating system macro in the SMARTS server environment.

**Placeholders**

\$1	Name of the SMARTS server system macro that failed.
\$2	Return code from the operation (normally register 15).
\$3	Feedback code from the operation (normally register 0).
\$4	Reserved for future use; currently the same as \$2.

**Action** Based on any preceding messages and the SMARTS server macro return and feedback codes, determine the cause of the problem and take action to correct it. If the action to take is not obvious, report the error to your Software AG technical support representative.

**References**

Refer to appendix G in this manual.

**PSX0048 Thread id \$1 Abended code X\$2/\$3**

**Description** A pthread ABENDED in the SMARTS server environment.

**Placeholders**

\$1	The pthread ID.
\$2	The ABEND code in hexadecimal.
\$3	The ABEND code in binary.

**Action** This message is provided as a diagnostics message: determine why the ABEND is occurring.

**References**

Not applicable.

**PSX0049 Required configuration parameter \$1 missing oR invalid**

**Description** During initialization of a CDI protocol driver, a required configuration parameter was missing or invalid. As a result, the initialization of the protocol driver cannot proceed.

**Placeholders**

\$1	Name of the missing or invalid configuration parameter.
-----	---

**Action** Add or correct the required configuration parameter and restart SMARTS.

**References**

SMARTS Installation and Operations Manual.

**PSX0050 CDI \$1 protocol initialized**

**Description** A CDI protocol driver implementing the protocol specified in \$1 has been successfully initialized.

**Placeholders**

\$1	Name of the CDI protocol initialized.
-----	---------------------------------------

**Action** Not applicable.

**References**

Not applicable.

**PSX0051 Too many arguments to fit in argv list**

**Description** The nucleus attempted to build an argv buffer based on the input provided by the user for a SMARTS spawn\*() or exec\*() request; however, there was insufficient room in the ARGB buffer to hold all of the data. The size of the ARGB buffer may be determined from the \_POSIX\_ARG\_MAX macro variable.

**Placeholders**

None

**Action** Reduce the amount of data being passed in the ARGB buffer.

**References**

SMARTS SDK Programmer's Reference Manual SMARTS SDK Programmer's Guide

**PSX0052    Function \$1 '\$2' Unsupported**

**Description** FUNCLIST=YES is specified in the SMARTS configuration. A message is issued for each POSIX function that is registered within SMARTS but for which support has not yet been implemented.

**Note:** The fact that the function has been registered within the SMARTS nucleus does not imply that it will be supported at any time in the future.

**Placeholders**

\$1	Name of the POSIX function.
\$2	A short description of what the POSIX function does.

**Action** No action required. If the message is no longer required, turn off the FUNCLIST configuration parameter.

**References**

Not applicable.

**PSX0053    Statistics Collection for: \$1**

**Description** Indicates the functions for which statistics are being collected.

**Placeholders**

\$1	List of all functions for which statistics are being collected. "NONE" if no statistics are being collected.
-----	--

**Action** This message is for information only. No action is required.

**References**

*SMARTS Installation and Operations Manual.*

**PSX0054    CDI \$1 protocol terminated**

**Description** A CDI protocol handler has been terminated. This normally only happens during termination of the SMARTS environment.

**Placeholders**

\$1	Name of the CDI protocol.
-----	---------------------------

**Action** No action is required.

**References**

Not applicable.

**PSX0055 CDI \$1 protocol initialization error: \$2**

**Description** An error occurred during CDI protocol handler initialization.

**Placeholders**

\$1	Name of the CDI protocol.
\$2	Return code from the CDI protocol handler.

**Action** This will normally be a parameter error. Correct and retry.

**References**

SMARTS Installation and Operations Manual.

**PSX0056 Failure to delete file '\$1'**

**Description** An error occurred in the processing of a file delete.

The file delete is attempted subsequently to a return value indicating success. This may occur if a file is open at the time of the delete command and the file is marked to be deleted at a later stage.

**Placeholders**

\$1	Name of the file that was not deleted.
-----	--

**Action** Check for messages in the log that may indicate the reason for failure, such as a security violation. The file should be deleted if required.

**References**

Not applicable.

**PSX0057 Pipe \$1 request error PID=\$2 RC=\$3 REASON=\$4**

**Description** An error has occurred while reading from or writing to a pipe. The placeholders indicate the action that encountered the error.

**Placeholders**

\$1	The file descriptor number of the pipe.
\$2	The ID of the process that encountered the error.
\$3	The return code from the pipe CDI protocol handler.
\$4	An internal reason code.

**Action** This error may be caused by other error situations in the environment, such as lack of memory. Look for other symptoms, correct and retry. If the problem persists, contact your Software AG technical support representative.

**References**

Not applicable.

**PSX0058 DXR Server \$1 \$2 \$3**

**Description** An error has occurred while dealing with a SMARTS-based server running in the SMARTS server environment. The placeholders indicate the action that encountered the error.

**Placeholders**

\$1	The name of the server.
\$2	an indication as to what the issue was: Initialization – Problems with initialization. Termination – Problems with termination. Command processing – Problems processing a command for this server. Not Active – The server is not active. No longer active – The server has terminated since the last activity.
\$3	'FAILED' or blanks, depending on the context of the message.

**Action** Clear up the problem leading to the message and retry the operation.

**References**

Not applicable.

**PSX0059 DXR server \$1 \$2 \$3 PID=\$4**

**Description** An operation on a SMARTS-based server running in the SMARTS server environment was completed successfully.

**Placeholders**

\$1	The name of the server.
\$2	An indication of what was successful: initialization, termination, or command processing.
\$4	The process ID for the named server.

**Action** No action required.

**References**

Not applicable.

**PSX0060 DXR \$1 function failed RC=\$2 RV=\$3 errno=\$4**

**Description** SMARTS itself issued a POSIX interface request that failed.

**Placeholders**

\$1	Name of the POSIX function.
\$2	The assembler interface return code received from the function.
\$3	Return value received from the function.
\$4	Errno returned from the function.

**Action** Determine the cause of the problem using the return value, errno, and the SMARTS SDK Programmer's Reference Manual . If the error is expected for some reason, ignore the message. If it should not have occurred, correct the problem leading to the error.

**References**

SMARTS SDK Programmer's Reference Manual SMARTS SDK Programmer's Guide

**PSX0062 Member \$1 not stored in \$2. No space left in directory**

**Description** No space remains in the directory: the file could not be added or the file name could not be changed.

**Placeholders**

\$1	File name.
\$2	Directory name.

**Action** Increase the directory or the directory index space as appropriate for the operating system file system.

**References**

Refer to the manuals specific to your operating system.

**PSX0063 Invalid keyword value specified for '\$1'**

**Description** An invalid keyword value was passed in by the configuration file. Initialization continues, ignoring the invalid value.

**Placeholders**

\$1	Name of the keyword that received an invalid value.
-----	---

**Action** Check the SMARTS Installation and Operations Manual for a valid keyword value.

**References**

SMARTS Installation and Operations Manual .

**PSX0064 Trace DataSpace InitialiZed, ESIZE=\$1:BSIZE=\$2:NBLKS=\$3**

**Description** Informational message issued at SMARTS initialization.

**Placeholders**

\$1	Trace data space element size.
\$2	Trace data space block size.
\$3	Number of blocks in the trace data space.

**Action** If different values are required, set them in the configuration file.

**References**

SMARTS data collection facilities document.

**PSX0065 Log DataSpace InitialiZed, ESIZE=\$1:BSIZE=\$2:NBLKS=\$3****Description** Informational message issued at SMARTS initialization.**Placeholders**

\$1	Log data space element size.
\$2	Log data space block size.
\$3	Number of blocks in the log data space.

**Action** If different values are required, set them in the configuration file.**References**

SMARTS data collection facilities document.

**PSX0066 Trace level = '\$1'****Description** Indicates the level of tracing in the system.**Placeholders**

\$1	Level of tracing between 1 and 5, '1' being the least amount of trace data and 5 being the greatest.
-----	--

**Action** This message is for information only. No action is required.**References**

SMARTS Installation and Operations Manual .

**PSX0067 System tracing enabled = '\$1'****Description** Indicates the system tracing that is turned on.**Placeholders**

\$1	List of system tracing that is turned on.
-----	---

**Action** This message is for information only. No action is required.**References**

SMARTS Installation and Operations Manual .

**PSX0068 No System Tracing enabled**

**Description** Indicates that no system tracing is enabled.

**Placeholders**

Not applicable.

**Action** This message is for information only. No action is required.

**References**

SMARTS Installation and Operations Manual .

**PSX0069 No functions are being traced**

**Description** Indicates that no system tracing is enabled.

**Placeholders**

Not applicable.

**Action** This message is for information only. No action is required.

**References**

SMARTS Installation and Operations Manual .

**PSX0070 All functions are being traced**

**Description** Indicates that all functions are being traced.

**Placeholders**

Not applicable.

**Action** This message is for information only. No action is required.

**References**

SMARTS Installation and Operations Manual .

**PSX0071 Functions being traced: '\$'**

**Description** Indicates the functions that are being traced.

**Placeholders**

\$1	List of all functions that are being traced.
-----	--

**Action** This message is for information only. No action is required.

**References**

SMARTS Installation and Operations Manual .

**PSX0072 Functions not being traced: '\$'**

**Description** Indicates the functions that are not being traced.

**Placeholders**

\$1	List of all functions that are not being traced.
-----	--

**Action** This message is for information only. No action is required.

**References**

SMARTS Installation and Operations Manual .

**PSX0073 P=\$1 T=\$2 \$3 at \$4+X\$5**

**Description** Back trace message.

**Placeholders**

\$1	Process ID
\$2	Thread ID
\$3	'Function call' or 'Abend'
\$4	Function name
\$5	Offset

**Action** This message is for information only. No action is required.

**References**

Not applicable.

**PSX0074** P=\$1 T=\$2 \$3:\$4 \$5 \$6 \$7 \*\$8\*

**Description** Back trace data.

**Placeholders**

\$1	Process ID	
\$2	Thread ID	
\$3	Storage address	
\$4	Storage contents - hex	
\$5	Storage contents - hex	
\$6	Storage contents - hex	
\$7	Storage contents - hex	
\$8	Storage contents - character	

**Action** This message is for information only. No action is required.

**References**

Not applicable

**PSX0075** P=\$1 T=\$2 aborted due to \$3 signal

**Description** Abort due to signal.

**Placeholders**

\$1	Process ID	
\$2	Thread ID	
\$3	Signal number	

**Action** This message is for information only. No action is required.

**References**

Not applicable

**PSX0076** Out of storage - increase region size

**Description** Error - increase region size.

**Placeholders**

None

**Action** Increase region size, rerun.

**References**

SMARTS Installation and Operations Manual.

**PSX0077 PFS termination checkpoint failed**

**Description** PFS checkpoint failure.

**Placeholders**

None

**Action** Report this message to your technical support representative.

**References**

Not applicable.

**PSX0078 PFS container capacity at \$1%**

**Description** PFS container is in danger of filling up.

**Placeholders**

\$1	Percentage used
-----	-----------------

**Action** Once this message has been received more than once for a particular container it should be resized

**References**

SMARTS Installation and Operations Manual.

**PSX0079 \$2 block \$1 corrupted**

**Description** A block was corrupted.

**Placeholders**

**Action** Report this message to your technical support representative.

**References**

Not applicable.

**PSX0080 Error in "\$1", record number \$2**

**Description** Invalid record in hosts, protocols, services or networks file.

**Placeholders**

\$1	Name of file containing invalid record.
\$2	Record number of invalid record in file.

**Action** Check relevant record in input file for error.

**References**

SMARTS Installation and Operations Manual.

**PSX0081 Error on PUT, DDNAME=\$1**

**Description** PUT error, Data collection file.

**Placeholders**

\$1	ddname
-----	--------

**Action** Report this message to your technical support representative.

**References**

SMARTS Installation and Operations Manual.

**PSX0082 ENQ/DEQ failure, DDNAME=\$1**

**Description** ENDQ/DEQ Data collection problem.

**Placeholders**

\$1	ddname
-----	--------

**Action** Report this message to your technical support representative.

**References**

SMARTS Installation and Operations Manual.

**PSX0083 Dataspace Max Size \$1 exceeded**

**Description** Maximum dataspace size exceeded.

**Placeholders**

\$1	Maximum size
-----	--------------

**Action** Report this message to your technical support representative.

**References**

SMARTS Installation and Operations Manual.

**PSX0084 PAeNDSM Module not available**

**Description** DSM module not available.

**Placeholders**

None

**Action** Report this message to your technical support representative.

**References**

SMARTS Installation and Operations Manual.

**PSX0085 Dataspace \$1 not initialised correctly**

**Description** The Dataspace was not successfully initialised.

**Placeholders**

\$1	Dataspace name
-----	----------------

**Action** Report this message to your technical support representative.

**References**

SMARTS Installation and Operations Manual.

**PSX0086 Dataspace \$1 not terminated correctly**

**Description** The Dataspace was not successfully terminated.

**Placeholders**

\$1	Dataspace name
-----	----------------

**Action** Report this message to your technical support representative.

**References**

SMARTS Installation and Operations Manual.

**PSX0087 CEEPIPI Interface module not available**

**Explanation** Unable to load the LE PIPI interface module, CEEPIPI

**Action** Make the module available to the job and rerun.

**References**

OS/390 LE Manuals.

**PSX0088 LE PIPI \$1 Error rf=\$2 rc=\$3 rsn=X\$4 fbc=X\$5**

**Explanation** An error has occurred while interfacing with the LE subsystem on OS/390. It indicates a possible installation problem for SMARTS.

**Placeholders**

\$1	LE Call
\$2	response code
\$3	return code
\$4	reason code
\$5	feedback code

**Action** Report this message to your technical support representative.

**References**

Relevant OS/390 LE Manuals.

**PSX0089 DXR/IMS: \$1 of token failed - \$2, return code \$3**

**Explanation** An MVS token creation, retrieval or deletion has failed.

**Placeholders**

\$1	failing request
\$2	token
\$3	return code

**Action** Report this message to your technical support representative.

**References**

IBM MVS Programming Assembler Services Reference

**PSX0090 Warning: Error in freeing of storage addr \$1**

**Explanation** An Error occurred during the freeing of storage - corruption detected

**Action** Contact Tech Support

**PSX0091 TCPIP System Id (SYSPARM) specified='\$1'**

**Explanation** Identifies the TCP/IP for VSE partition that the SMARTS Sockets interface has connected with.

**Placeholders**

\$1	This was indicated by the //OPTION SYSPARM='nn' statement in the application startup JCL. The ID assignment is made in the parameter field of the EXEC statement of the TCP/IP for VSE startup JCL.
-----	---

**Action** This message is for information only.

**References**

VSE/ESA V2R6 System Control Statements

**PSX0092 Error freeing storage for \$1 (address \$2)**

**Explanation** An Error occurred during the freeing of a DXR Control Block at Address \$2

**Action** Contact Tech Support with the text of this message

**PSX0094 Sockets BUFSZ=\$1, MINQ=\$2, MAXQ=\$3**

**Explanation** Indicates the values used (explicitly or by default) by the Sockets subsystem.

**Placeholders**

\$1	The size of the input and output buffers.
\$2	The minimum listen() queue length or backlog that can be specified by an application.
\$3	The maximum listen() queue length or backlog that can be specified by an application.

**Action** This message is for information only.

**References**

Not applicable

**PSX0095 Sockets Port \$1 listening. Queue length \$2**

**Explanation** A server has opened a socket and is waiting for work.

**Placeholders**

\$1	The port number of the server.
\$2	The listen() queue length or backlog that the Sockets subsystem will use for the socket. This has been specified by the server application or has been forced by default.

**Action** This message is for information only.

**References**

Not applicable

**PSX0096 Sockets Port \$1 closed. Connections \$2, Concurrent \$3**

**Explanation** Indicates that a previously listening port has been closed.

**Placeholders**

\$1	The port number of the server.
\$2	The total number of connections made with the server port.
\$3	The highest number of concurrent connections made.

**Action** This message is for information only.

**References**

Not applicable

---

**PSX0099**    \$1 \$2 \$3 \$4 \$5 \$6 \$7 \$8

**Explanation** Trace data.

**Placeholders**

\$1-\$8	Variable information.
---------	-----------------------

**Action**        This message is for information only.

Report this message to your technical support representative.

**References**

Not applicable



# 5

## SMARTS Server Environment Messages

---

▪ ABE — ABEND Processing .....	50
▪ ABS — Service Routine Processing .....	53
▪ ADA — ADABAS Interface .....	57
▪ BPM — Buffer Pool Management .....	59
▪ DIS — Dispatching Mechanism Messages .....	63
▪ INI — Initialization : Main Processing .....	71
▪ INP — Initialization : Parameter Processing .....	75
▪ LOD — Program Management Services .....	76
▪ OPC — Operator Communication .....	78
▪ OUS — ";User" Operator Command Processor .....	96
▪ RES — Reentrant Program Support .....	97
▪ ROL — Rollout / Rollin Processing .....	100
▪ RSM — Resource Management .....	103
▪ STG — Storage Initialization .....	104
▪ SVR — Server Processing .....	106
▪ TIB — Terminal Initialization .....	108
▪ TMR — Timing Services .....	109
▪ ZDM — COMDMP Dump Dataset Processing(VSE Only) .....	114
▪ ZLA — Security and Accounting .....	117
▪ ZOS — OS Initialization .....	117
▪ ZTR — Trace Utilities .....	119
▪ ZTS — Thread Storage .....	119

This document contains descriptions of the messages returned within the SMARTS Server environment:

## ABE — ABEND Processing

---

**ABE0001**      \$1 - \$2 \$3 \$4 \$5 \*\$6\*

**Explanation**      Prior to a system dump being taken, this message is printed once for every allocated 16 bytes of storage around the ABEND PSW and register set. It consists of the address of the storage being printed, followed by the 16 bytes of information pointed to by the address in hex and character formats.

**Sys. Programmer** These messages can be of interest to your support personnel when you are reporting a problem. They should be kept as a part of a diagnostic summary of an ABEND unless the problem is known and diagnostics are no longer required for the problem in question.

**Comp. Operator** This error message is normally issued as part of a diagnostic set of ABEND messages from the SMARTS server environment. Keep these and any other ABE messages from the job and pass them to the systems programmer.

**ABE0002**      SMARTS \$1 task ABEND system \$2 user \$3

**Explanation**      A SMARTS server environment subtask ABENDED with the indicated system and/or user ABEND code. This message is followed by one or more ABE messages with diagnostic data for the ABEND.

**Sys. Programmer** This is normally the first message from the SMARTS server environment nucleus when a SMARTS server environment ABEND occurs for which a dump is taken. It is followed by one or more ABE messages containing diagnostic information from the ABEND that will be required if the error is reported as a problem to your technical support representative.

**Comp. Operator** Inform your systems programmer about the message.

**ABE0003**      SDWA not available for printing

**Explanation**      During ABEND processing, the SMARTS server environment abnormal termination/recovery routines determined that the MVS SDWA control block, normally available during ABEND processing, was not available for this particular ABEND. In this case, only a minimum of diagnostic information can be printed.

**System Action**      The SMARTS server environment continues processing according to the other ABE messages printed.

**Sys. Programmer** For some reason, the IBM recovery routines could not build an SDWA for a SMARTS server environment ABEND. This normally indicates a shortage of storage in the SMARTS server environment region but this does not necessarily have to be the case. Refer to the various IBM publications as to the possibilities why the SMARTS server environment could not be provided with a SDWA and correct the situation.

**Comp. Operator** Inform your systems programmer about the message.

**ABE0004 Dump will be taken for \$1 task**

**Explanation** A SMARTS server environment ABEND has occurred and the SMARTS server environment has determined that a dump should be taken for the particular SMARTS server environment task.

**Sys. Programmer** This indicates that the SMARTS server environment will attempt to take a dump for the subtask mentioned in the message using recovery or termination processing.

**Comp. Operator** Inform your systems programmer.

**ABE0005 Recovery in progress for \$1 task**

**Explanation** A SMARTS server environment subtask abended. The SMARTS server environment has determined that recovery may be possible and therefore will be attempted. If this is preceded by ZAB00004, then the recovery may take some time due to the fact that the dump must be taken before recovery can actually take place.

**Sys. Programmer** Under normal circumstances, recovery is always attempted unless a recursive loop has occurred. If this is detected, no other recovery is attempted. If it is not detected, the results are unpredictable. In some cases, it may be necessary to use the operating system functions to terminate the SMARTS server environment; however, this should never occur.

**ABE0006 Abend PSW \$1 \$2 \$3 \$4**

**Explanation** This message contains more diagnostic information about the SMARTS server environment abend that occurred. The first two fullwords contain the PSW for the abend. The second two fullwords contain the 'PSW2' contents at the time of the abend.

**ABE0007 R0=\$1 R1=\$2 R2=\$3 R3=\$4**

**Explanation** Abend diagnostics showing the contents of registers 0 (zero) to 3 (three) at the time of a SMARTS server environment abend.

**ABE0008 R4=\$1 R5=\$2 R6=\$3 R7=\$4**

**Explanation** Abend diagnostics showing the contents of registers 4 (four) to 7 (seven) at the time of a SMARTS server environment abend.

**ABE0009 R8=\$1 R9=\$2 RA=\$3 RB=\$4**

**Explanation** Abend diagnostics showing the contents of registers 8 (eight) to b (eleven) at the time of a SMARTS server environment abend.

**ABE0010** RC=\$1 RD=\$2 RE=\$3 RF=\$4

**Explanation** Abend diagnostics showing the contents of registers c (twelve) to f (fifteen) at the time of a SMARTS server environment abend.

**ABE0011** \$1 - Address not accessible

**Explanation** While attempting to print the storage areas around the SMARTS server environment PSW and register set, the storage location in the message was found to be either not allocated or paged out at the time the abend was taken. For this reason, the storage could not be printed.

**Sys. Programmer** This is not an error: the message simply indicates that the storage could not be addressed during dump processing. If the storage exists and was simply paged out, it will still appear in the actual dump if a dump is taken.

**ABE0012** \$1 failed during dump processing, reason code ` \$2`

**Explanation** An error occurred in the SMARTS server environment, causing the recovery routines to attempt to create a dump; however, the \$1 processing for this failed. The reason code \$2 indicates why the function failed.

Possible values for \$1:

GETSTOR	Storage could not be obtained from the general buffer pool. In this case, \$2 will contain the internal return code.
DYNALLOC	Dynamic allocation for the output dataset failed. The error and reason codes from the dynalloc will be returned as the reason code.
OPEN	The open for the dataset failed. In this case, \$2 will contain zero.

**System Action** The system will continue processing if recovery is successful.

**Sys. Programmer** Using the reason code and any system messages, try to establish the reason for the failure. If the error is system-dependent, correct the error. If the error appears to be software-dependent, contact your technical support representative.

**ABE0013** Logic error in program \$1 at +x'\$2' TID \$3 LUname \$4

**Explanation** A logic error occurred processing TID \$1 with luname \$2. This message will be followed by one or more ZAB messages that dump diagnostic information to the console.

**System Action** Processing continues; however, the TID in question may be lost. It may be recoverable using the LOGOFF and/or FORCE operator commands.

**Sys. Programmer** This indicates a logic error within the SMARTS server environment nucleus. Provide any messages before this message plus the diagnostics that are issued after this message to your technical support representative.

ABE0014 \$1

ABE0015 **Dump suppressed**

**Explanation** Same error message as previous dump.

## ABS — Service Routine Processing

---

ABS0001 **\$1 cancelled due to invalid parameter**

**Explanation** Program \$1 was cancelled as it attempted to invoke a SMARTS server environment function with a parameter list that did not exist in storage.

**System Action** The program abends with a dump and the message is sent to the user.

**Term. Operator** Report this application program error to the applicable person.

**Appl. Programmer** A dump is produced from which you can determine the point in the program at which the invalid parameter or parameter list was passed to the SMARTS server environment. Correct the parameter list error and rerun the program.

ABS0002 **Program cancelled by terminal operator**

**Explanation** This SMARTS server environment service routines message appears when the terminal user issues the SMARTS server environment cancel command ('\*CANCEL') from the terminal.

**System Action** The program is cancelled.

**Term. Operator** For information only; no action is required.

ABS0003 **Program \$1 cancelled by \$2 operator**

**Explanation** This SMARTS service environment service routines message indicates that the computer operator or a control terminal user issued the SMARTS server environment `CAN' command to cancel a terminal program.

**System Action** The program is cancelled.

**Term. Operator** For information only. Contact the computer operator to determine why the program was cancelled.

**Appl. Programmer** Determine the cause for the cancel request. Correct any errors and reexecute the program.

<b>ABS0004</b>	<b>\$1 cancelled due to invalid parameter list address</b>
<b>Explanation</b>	This SMARTS server environment service routines message indicates that the application program was abnormally terminated by the SMARTS server environment because the address of the parameter list was invalid. This normally means that the parameter list was not boundary-aligned.
<b>System Action</b>	Program \$ is cancelled.
<b>Term. Operator</b>	Contact the application programmer responsible for the program in use when the error occurred.
<b>Appl. Programmer</b>	Determine the cause for the invalid parameter address. The invalid parameter address is in register 1.
<b>ABS0005</b>	<b>\$1 tried to execute invalid SVC / function X'\$2'</b>
<b>Explanation</b>	This SMARTS server environment service routines message indicates that a SMARTS server environment service was requested by a user program and the function code for the service was invalid. The invalid function code is expressed as the hexadecimal number X'\$2'.  This message is also issued if an invalid address has been specified for an OS SVC parameter.
<b>System Action</b>	The program '\$1' is abnormally terminated with a SMARTS server environment online dump.
<b>Term. Operator</b>	This error was probably caused by the application program. Notify the application programmer responsible for program '\$1'.
<b>Sys. Programmer</b>	The PSW contains the address of the location, which follows both the invalid function code and the SVC. The SVC requests SMARTS server environment services.
<b>Appl. Programmer</b>	The condition that causes ZSR00005 can occur when a nonprivileged user program executes a privileged request or inadvertently branches to a random location, or when the area of a user program that contained a valid function code for the SMARTS server environment was destroyed before the request for SMARTS server environment services was executed. SMARTS server environment function codes occur in MCALL macro expansions in BAL programs or in subroutines for SMARTS server environment functions in programs written in higher level languages.
<b>ABS0006</b>	<b>PROGRAM \$1 Abend S\$2 PSW=\$3</b>
<b>Explanation</b>	This SMARTS server environment service routines message indicates that the program '\$1' was abnormally terminated by the resident operating system (OS).
<b>System Action</b>	Program '\$1' is abnormally terminated with a SMARTS server environment online dump.
<b>Term. Operator</b>	This error was probably caused by the application program. Notify the application programmer responsible for program '\$1'.
<b>Appl. Programmer</b>	The IBM completion code associated with the termination is indicated by \$2. Refer to the appropriate hardware manual for information about this code. The PSW associated with the termination is indicated by \$3.

<b>ABS0007</b>	<b>PROGRAM \$1 Abend U\$2</b>
<b>Explanation</b>	This SMARTS server environment service routines message indicates that the program '\$1' was terminated at its own request for the reason indicated by the reason code \$2.
<b>System Action</b>	Program '\$1' is abnormally terminated with a SMARTS server environment online dump.
<b>Term. Operator</b>	Refer to the operating instructions for program '\$1' to determine the meaning of the code \$2. If no such instructions exist, seek the assistance of the application programmer responsible for program '\$1'.
<b>ABS0008</b>	<b>Program too long; not enough room to build save</b>
<b>Explanation</b>	This SMARTS server environment service routines message indicates that insufficient space remained in the thread for the required 18-word save when the SMARTS server environment attempted to load the user program.
<b>System Action</b>	The user program is abnormally terminated with a SMARTS server environment online dump.
<b>Term. Operator</b>	This error was probably caused by the application program. Notify the application programmer responsible for the program in use when the error occurred.
<b>Appl. Programmer</b>	Either decrease the size of the program or recatalog it with a larger region size.
<b>ABS0009</b>	<b>Program \$1 linked with planned overlay -</b>
<b>Explanation</b>	This SMARTS server environment service routines message indicates that an invalid overlay structure was found in program '\$1' while attempting to load the program.
<b>System Action</b>	The user program is abnormally terminated with a SMARTS server environment online dump.
<b>Term. Operator</b>	This error was probably caused by the application program. Notify the application programmer responsible for the program in use when the error occurred.
<b>Appl. Programmer</b>	Either decrease the size of the program or recatalog it with a larger region size. If this fails, examine the overlay structure for inconsistencies and relink the program before recataloging.
<b>ABS0010</b>	<b>Program \$1 requested function or SVC not allowed \$2</b>
<b>Explanation</b>	This SMARTS server environment service routines message indicates that the user program called a SMARTS server environment function or issued an SVC instruction that was not available for use by online programs.
<b>System Action</b>	The user program is abnormally terminated with a SMARTS server environment online dump.
<b>Term. Operator</b>	This error was probably caused by the application program. Notify the application programmer responsible for the program in use when the error occurred.
<b>Appl. Programmer</b>	A user program inadvertently branched to a bad location or the area of a user program that contained a valid function code for the SMARTS server environment was destroyed before the request for SMARTS server environment services was executed. SMARTS server environment function codes occur in MCALL macro expansions in BAL programs

or in subroutines for SMARTS server environment functions in programs written in higher level languages. The PSW contains the address of the location that follows both the invalid function code and the SVC. The SVC requests the service from the SMARTS server environment.

**ABS0011      Disk error loading \$1**

**Explanation**      This SMARTS server environment service routines message indicates that a disk error occurred while the SMARTS server environment was attempting to load program '\$1' into main storage.

**System Action**      Program '\$1' is not loaded. The application program making the load request is abnormally terminated and an online dump is taken.

**Term. Operator**      This error was probably caused by a hardware failure. Retry the operation. If the error continues to appear, contact the application programmer responsible for the program in use when the error occurred.

**Sys. Programmer**      The program library may require reinitialization and reloading. If the program being loaded is in a load library, ensure that the load library is not in secondary extents. If the error persists, move the program library and/or load library to another location.

**Appl. Programmer**      Recatalog program '\$1' and retry the operation. If the error persists, contact the SMARTS server environment system programmer.

**ABS0014      \$1 is security protected from calling terminal or**

**Explanation**      This SMARTS server environment service routines message indicates that the terminal operator made an unauthorized attempt to use the program '\$1'.

**System Action**      The request is ignored.

**Term. Operator**      The program '\$1' may not be used from your terminal. Contact the application programmer responsible for the indicated program.

**ABS0016      \$1 invalid function after internal '\*CANCEL', Tid \$2**

**Explanation**      A '\*CANCEL' may be issued internally for the following reasons:

1. The terminal on which a conversational program was running experienced a 'forced' logoff.
2. A stacked user program was killed using a minus ('-') from the USTACK map.

The terminal receives a return code to the conversational write or 'FLIP' request indicating that a '\*CANCEL' has occurred.

The user program is given control to clean up if necessary and then should issue an MCALL WRT with the `done' option, an MCALL EOJ, or an MCALL ABEND. In these cases for an internal `\*CANCEL' the request is simply treated as an EOJ.

If following an internal `\*CANCEL' the user program abends, issues a MCALL WRT without the `done' option, or requests a roll out via any means, this message is issued and the request is then treated as an EOJ.

**System Action** In all cases the program environment is successfully cleaned up.

**Comp. Operator** Report this message to the person responsible for the program so that the '\*CANCEL' processing in the program can be corrected.

## ADA — ADABAS Interface

**ADA0001**      **Program \$1 cancelled - invalid address in or for**

**Explanation** One or more of the following errors were the program's (named \$1) ADABAS parameter list or ACB:

•	The location specified to contain the ADABAS parameter list is not within the area available to the application program.
•	One or more of the locations specified to contain the ADABAS control block (ACB) or ADABAS buffers is not within the area available to the application program.
•	One or more of the lengths specified for the ADABAS buffers is too large or negative.

**System Action** The application program is abnormally terminated and a dump is taken.

**Term. Operator** This error was caused by the application program. Contact the programmer responsible for the program in use when the error occurred.

**Appl. Programmer** Register 1 in the dump contains the address the program supplied for the ADABAS parameter list. Check this address, the buffer addresses, and the buffer lengths to determine which addresses or lengths are invalid.

**ADA0002**      **Program \$1 cancelled - no space in buffer pool**

**Explanation** This SMARTS server environment ADABAS interface message indicates that SMARTS server environment was unable to successfully perform an ADABAS call for program \$1 because there was insufficient room in the general buffer pool for the ADABAS buffers required by the interface.

**System Action** The application program is abnormally terminated and a dump is taken.

If the program issues an ADABAS call that requires an extremely large amount of data to be transferred, it may be possible to modify it to make more calls transferring a smaller amount of data with each call.

If this modification cannot be made or if the amount of data being transferred is not considered to be excessive, consult the SMARTS server environment system programmer about increasing the size of the SMARTS server environment's region to allow for general buffer pool expansion.

**Term. Operator** The condition causing the error may be temporary and due to heavy use of ADABAS. Wait a few minutes and try again. If the problem persists, report the problem to the systems programmer responsible as it indicates that insufficient space has been allocated at startup of the SMARTS server environment.

If the program issues an ADABAS call that requires an extremely large amount of data to be transferred, it may be possible to modify it to make more calls transferring a smaller amount of data with each call.

If this modification cannot be made or if the amount of data being transferred is not considered to be excessive, consult the SMARTS server environment system programmer about increasing the size of the SMARTS server environment's region to allow for general buffer pool expansion.

**Sys. Programmer** As this buffer is acquired from the SMARTS server environment's ADABAS buffer pool, this message indicates that there is insufficient buffers available at certain times to satisfy all requests.

If the program issues an ADABAS call that requires an extremely large amount of data to be transferred, it may be possible to modify it to make more calls transferring a smaller amount of data with each call.

If this modification cannot be made or if the amount of data being transferred is not considered to be excessive, consult the SMARTS server environment system programmer about increasing the size of the SMARTS server environment's region to allow for general buffer pool expansion.

**Appl. Programmer** Register 1 in the dump contains the address of the ADABAS parameter list, which contains the address of the ADABAS control block (ACB) and buffers. From this information, the ADABAS command can be determined and the amount of data being transferred to and/or from ADABAS can be verified.

If the program issues an ADABAS call that requires an extremely large amount of data to be transferred, it may be possible to modify it to make more calls transferring a smaller amount of data with each call.

If this modification cannot be made or if the amount of data being transferred is not considered to be excessive, consult the SMARTS server environment system programmer about increasing the size of the SMARTS server environment's region to allow for general buffer pool expansion.

**ADA0003 Program \$1 cancelled - ADABAS calls exceeded**

**Explanation** This SMARTS server environment ADABAS interface message indicates that there were more ADABAS calls than specified in the ADALIMIT start-up parameter after the last terminal I/O.

**System Action** The application program is abnormally terminated and a dump is taken.

**Term. Operator** This error was caused by the application program. Contact the programmer responsible for the program in use when the error occurred.

**Sys. Programmer** Check the size specified for the ADALIMIT start-up parameter.

**Appl. Programmer** This problem can arise if either the ADALIMIT parameter is too small or the application program does too much or too complex work in one dialog. Other reasons include an increased amount of data within ADABAS or a loop in the application program.

## BPM — Buffer Pool Management

---

- BPM0001**      **BP \$1, ADDR=\$2 not in buffer pool ret=\$3**
- Explanation**      An attempt was made to free the fixed buffer pool element address \$2; however, this address is not allocated in the \$1 buffer pool. The request was issued from the location indicated by \$3.
- Sys. Programmer**      An invalid free request was issued for the buffer pool as indicated by \$1. Using the \$3 address, determine the module and offset from which the request was issued. The message generally indicates a problem with the usage of buffer pool \$1. When buffer pool \$1 is created by Software AG, report this to your technical support representative.
- 
- BPM0002**      **BP \$1 SP \$2(\$3), ADDR=\$4 bndry error ret=\$5**
- Explanation**      An attempt was made to free the fixed buffer element address \$4. This buffer was found to be within the buffer pool \$1 and the subpool as identified by \$2 and \$3; however, the address provided did not point to the start of a buffer in this subpool. The request was issued from the location indicated by \$5.
- Sys. Programmer**      An invalid free request was issued for the buffer pool as indicated by \$1. Using the \$5 address, determine the module and offset from which the request was issued. This message indicates a problem with the usage of buffer pool \$1. When buffer pool \$1 is created by Software AG, report this to your technical support representative.
- 
- BPM0003**      **BP \$1 SP \$2(\$3), ADDR=\$4 already free ret=\$5**
- Explanation**      An attempt was made to free the fixed buffer address \$4 in the \$1 fixed buffer pool. The address was found to be in subpool name \$2 ID \$3; however, it was already free. The request was issued from the location indicated by \$5.
- Sys. Programmer**      A module twice attempted to free the buffer identified by \$4. Using the \$3 address, determine the module and offset from which the request was issued. This message highlights a logic error with the usage of the \$1 buffer pool. When buffer pool \$1 is created by Software AG, report this to your technical support representative.
- 
- BPM0004**      **BP \$1 SP \$2(\$3), Expansion about to occur**
- Explanation**      A 'get' request has been issued for the \$1 buffer pool and can be resolved by the subpool \$2 ID \$3. This subpool and any extensions that may have previously been allocated is full and thus another extension must be built.
- System Action**      An attempt is made to create the extensions. A subsequent message indicates the success or otherwise the attempt to expand.
- Sys. Programmer**      When this occurs frequently for the same subpool, consider increasing the base allocation for the subpool to avoid the overhead of expansion.

**BPM0005 BP \$1 SP \$2(\$3), Expansion failed, status=\$4**

**Explanation** An attempt to expand subpool \$2 ID \$3 in the \$1 buffer pool failed. \$4 contains the status of the request in hexadecimal format. This represents a two-byte return code followed by a two-byte feedback code indicating why the expansion request failed. See section *Request Status Codes* of this document for detailed information.

**Sys. Programmer** Determine why the expansion failed based on the status as indicated by \$4. This generally only occurs due to a shortage of storage in the region. In this case, review the size of the region in which the SMARTS server environment is running or reduce the usage of the storage that is in short supply.

**BPM0006 SP \$1(\$2) Esize=\$3 Eno=\$4 Size=\$5 Loc=\$6 Key=\$7**

**Explanation** A new subpool or subpool extension is allocated by the fixed buffer pool manager. A preceding or subsequent message indicates why it has been allocated. Placeholder values are as follows:

\$1	Subpool name
\$2	Numeric subpool ID
\$3	The element size contained in this subpool in bytes
\$4	The number of elements allocated in this subpool
\$5	The total size of storage allocated for this subpool
\$6	Where the subpool storage resides: ANY BELOW DS ECSA CSA
\$7	The storage protect key that the subpool storage has assigned. This is normally the SMARTS server environment's key.

**BPM0007 BP \$1, Creating SP Esize=\$2 Opt=\$3 Ret=\$4**

**Explanation** A 'Get' request was issued for the \$1 buffer pool; however, no subpool exists to satisfy the request. As the buffer pool was created with an option indicating that the subpools should automatically be created if no match is found, the fixed buffer pool manager is about to attempt to create a subpool to match the request. The buffer subpool will be built with an element size of \$2 and with options \$3. These options are the hexadecimal option bytes as passed to the 'Get' request and are described in the CMFBPM macro. \$4 is the address from where the 'Get' request was issued.

**System Action** Additional messages are issued indicating the success or otherwise the attempt to create the new subpool.

**BPM0008 BP \$1, Create failed status=\$2**

**Explanation** An attempt to create a new buffer subpool for the \$1 buffer pool failed. \$2 is the status indicating the reason for the failure. This status is a hexadecimal representation of a return and feedback code with the first two bytes representing the return code and the second two bytes the feedback code. See appendix H, Request Status , starting on page for detailed information.

**System Action** The program that issued the 'Get' request which resulted in an attempt to create a subpool to satisfy the request will be notified that the 'Get' failed.

**BPM0009 BP \$1 SP \$2(\$3), \$4 expansion(s) contracted**

**Explanation** The subpool \$2 ID \$3 in the buffer pool \$1 was previously expanded due to excessive demands on the space allocated in the base area. The fixed buffer pool manager has determined that enough space now exists to delete some of the expansions. In this case, \$4 indicates the number of expansions that have been deleted and are no longer available to the subpool. They can of course be allocated again in the future if required.

**BPM0010 BP \$1 SP \$2(\$3), A=\$4 token error '\$5'/'\$6' ret=\$7**

**Explanation** An attempt was made to free a fixed buffer element from the buffer pool \$1. This was found to be in subpool \$2 ID \$3 and to have been acquired with a token provided on the 'Get' request. On the free request, either no token was provided or a token was provided that did not match the token provided on the 'Get' request. The free request was issued from the location indicated by \$7. \$5 is the token that was provided on the 'Get' request and \$6 is the token that was provided on this request. If either are blank, no token was provided for the appropriate request. \$4 is the address of the token that was the target of the free request.

**System Action** The buffer is not freed.

**Sys. Programmer** An error occurred in the handling of the \$1 buffer pool. When a buffer is acquired with a token specified, the free request must be issued with the same token name before the free request will be processed. Using the \$6 address, determine the module and offset from which the request was issued. When buffer pool \$1 is created by Software AG, report this to your technical support representative.

**BPM0011 BP \$1 SP \$2(\$3), A=\$4 chain error \$5/\$6 ret=\$7**

**Explanation** A request was issued to free fixed buffer element from the \$1 buffer pool. The buffer was found in subpool \$2 ID \$3 and was acquired with a chain specified. In this case, the free request was issued without a chain base specified or the buffer element was not found on the chain provided. The request was issued from the location indicated by \$7. \$5 is the address of the SPDS for the buffer pool element and \$6 is the address of the chain base provided on the free request. \$4 is the address of the buffer that was the target of the free request.

**System Action** The buffer is not freed.

**Sys. Programmer** When a buffer is acquired with a chain request, the fixed buffer pool manager chains information for the buffer into the provided chain. To ensure system integrity, the buffer

must be removed from the chain before being freed. In this case either the free request did not provide a chain base or the buffer does not exist in the provided chain and therefore cannot be removed from the chain. A logic error exists in the use of the \$1 buffer pool. Using the \$6 address, determine the module and offset from which the request was issued. When buffer pool \$1 is created by Software AG, report this to your technical support representative.

**BPM0012 BP \$1, Subpool creation successful**

**Explanation** An attempt to create a new buffer subpool for buffer pool \$1 was successful. This is preceded by message `6' indicating the size and attributes of the newly created subpool.

**BPM0013 BP \$1 SP \$2(\$3), Expanded successfully**

**Explanation** An attempt to expand subpool \$2 ID \$3 in the \$1 buffer pool was successful. A preceding message indicates the attributes of the newly created subpool extension.

**BPM0014 BP \$1, Internal request returned status=\$2**

**Explanation** An internal request was issued to perform a fixed buffer pool management function for the \$1 buffer pool. The request completed successfully; however, the status information as indicated by \$2 was returned. \$2 is the hex representation of a two-byte return code and a two-byte feedback code. The first two bytes are the return code and should always be x'0004', while the second two bytes are the feedback code. See appendix H, Request Status , starting on page for detailed information. The request being issued was indicated by a preceding message.

**System Action** Processing continues. Normally, a message follows indicating that the indicated processing completed successfully.

**Sys. Programmer** This rarely issued message indicates the possibility of a logic error in the fixed buffer pool manager. Although no direct problems are associated with this message, report it to your technical support representative.

**BPM0015 BP \$1, Freeall request; Buffers already free ret=\$2**

**Explanation** A FREEALL request was issued for the \$1 buffer pool to free a group of buffers with certain attributes; however, this has resulted in a free request for a buffer that has already been freed. The request was issued from the address indicated by \$2.

**System Action** The FREEALL request is terminated, possibly resulting in buffers being left allocated when they are no longer in use.

**Sys. Programmer** A logic error exists in the handling of the \$1 buffer pool. Using the \$2 address, determine the module and offset from which the request was issued. When the buffer pool is controlled by Software AG, report this message to your technical support representative.

**BPM0016 BP \$1 Allocated successfully**

**Explanation** The buffer pool \$1 has been allocated successfully. One or more fixed buffer pool number `6' messages will be subsequently issued describing the various subpools created for this buffer pool.

**BPM0017 BP \$1 Deleted successfully**

**Explanation** The buffer pool \$1 has been successfully deleted.

**BPM0018 BP \$1 Request \$2 status=\$3**

**Explanation** A \$2 request for buffer pool \$1 returned a status \$3 to the caller. The caller indicated that a message should be issued in the event of a status for the request and this message is the result. The \$3 status consists of a halfword return code and halfword feedback code. See appendix H, Request Status , starting on page for detailed information. If the buffer pool no longer exists, \$1 contains the string '\$UNAVAIL'.

**System Action** The system continues as normally as possible. This message is generally only issued when a program is not in a position to handle a failure due to the nature of the module. For example, if the routine that builds output messages cannot acquire a buffer, it is unlikely that it will be able to acquire a buffer to print a message indicating that it cannot acquire a buffer.

**Sys. Programmer** When this occurs for a buffer pool created by Software AG, report it to your technical support representative.

## DIS — Dispatching Mechanism Messages

---

**DIS0001 Thread group \$1 added successfully**

**Explanation** The thread group \$1 has been added successfully and is available for use. Additional messages are issued indicating the subgroups that the thread group contains.

**DIS0002 Thread group \$1 modified**

**Explanation** The thread group \$1 has been modified successfully. Subsequent messages indicate the new make up of the thread group. Note that when thread subgroups and threads are deleted as a result of a modification, the thread subgroups and thread resources are only cleaned up once they have been quiesced; that is, when they are no longer in use.

**DIS0003 Thread group \$1 quiescing, waiting for \$2 users**

**Explanation** A request has been issued to delete thread group \$1; however, it must first be quiesced (that is, all users using the thread group must first terminate). The thread group is waiting on \$2 users to finish using the thread group before it can be deleted.

**System Action** At system termination, if the EOJ is not a forced EOJ, the system waits until all users are finished using the thread group. If the EOJ is forced, the message is issued but the fact that the thread group has not been successfully quiesced is ignored and termination processing continues.

**Sys. Programmer** If this message continually appears with the same number of users, it indicates that some users did not terminate correctly. Report this to your technical support representative, which may ask that you take a dump of the situation and send it to them for diagnosis.

**Comp. Operator** If this message appears continually with the same number of users, it indicates that some users may have terminated without removing their associated use count from the thread group. In this case, you may terminate the system by issuing a forced EOJ or, where a diagnostic dump is required, you may cancel the system with a dump.

**DIS0004 Thread group \$1 deleted successfully**

**Explanation** The thread group \$1 has been successfully quiesced and the resources associated with it have been freed.

**DIS0005 Thread group \$1 add failed rc=\$2 fdbk=\$3**

**Explanation** An attempt to add the thread group \$1 failed due to internal response code \$2 and feedback code \$3.

**System Action** The thread group is not added.

RC	FB	Reason
08	20	Changes already in progress; retry the operation
08	32	Insufficient storage for control blocks (in gen. buffer pool)
08	40	Insufficient storage for threads (in region/partition)

If any other combination appears, report the problem to your technical support representative.

**Sys. Programmer** This may occur if insufficient resources are available to allocate the thread group. The following lists the return code/feedback code combinations that may legitimately occur and their cause:

RC	FB	Reason
08	20	Changes already in progress; retry the operation
08	32	Insufficient storage for control blocks (in gen. buffer pool)
08	40	Insufficient storage for threads (in region/partition)

If any other combination appears, report the problem to your technical support representative.

**DIS0006 Thread group \$1 modify failed rc=\$2 fdbk=\$3**

**Explanation** An attempt to modify the thread group \$1 failed due the the return code \$2 and the feedback code \$3.

**System Action** Depending on the point in the processing where the error occurred, some of the modifications requested may have succeeded. The status of the thread group should be checked with the UCTRL online utility.

RC	FB	Reason
08	20	Changes already in progress; retry the operation
08	32	Insufficient storage for control blocks (in gen. buffer pool)
08	40	Insufficient storage for threads (in region/partition)

If any other combination appears, report the problem to your technical support representative.

**Sys. Programmer** This error can occur if insufficient resources are available in the system. The following lists the return code/feedback code combinations that may legitimately occur and their cause:

RC	FB	Reason
08	20	Changes already in progress; retry the operation
08	32	Insufficient storage for control blocks (in gen. buffer pool)
08	40	Insufficient storage for threads (in region/partition)

If any other combination appears, report the problem to your technical support representative.

**DIS0007 Subgroup name \$1 below=\$2 above=\$3 threads=\$4**

**Explanation** A thread group has been successfully added or modified. The variables are as follows:

\$1	The name of the thread subgroup
\$2	The amount of storage in the subgroup's threads below the line
\$3	The amount of storage in the subgroup's threads above the line
\$4	The number of threads in the subgroup
\$5	The key of those threads ('M' indicates mixed keys used)

**DIS0008      Program \$1 thread group \$2 not found**

**Explanation**    An attempt to start program \$1 failed because thread group \$2, which was explicitly allocated in the program's catalog entry, was not defined.

**Term. Operator** If the program should be available, report the error to your help desk.

**DIS0009      Program \$1 thread group \$2 quiescing**

**Explanation**    Program \$1 has been cataloged to run in thread group \$2; however, this thread group is no longer available in the system as it is quiescing.

**Term. Operator** If the program should be available, report the error to your help desk.

**DIS0010      Program \$1 no suitable thread subgroup**

**Explanation**    Program \$1 did not find a suitable thread subgroup within its thread group where it can run.

**Term. Operator** If the program should be available, report the error to your technical support representative.

**Sys. Programmer** Program \$1 found its thread group and attempted to find a subgroup with a thread size below the line sufficient to run it. It could not find such a subgroup and therefore could not run. Either the catalog size for the program must be reduced or a thread subgroup defined for the program's thread group must be large enough to run this program.

**DIS0011      Unexpected CMTHCM error pgm=\$1 tg=\$2 rc=\$3 fdbk=\$4**

**Explanation**    An error occurred during the dispatching cycle related to an internal CMTHCM request. The program that experienced the error is \$1; the thread group is \$2; and the return and feedback codes are \$3 and \$4, respectively.

**System Action**    The program will not start if it is an initialization request or will be terminated if the error occurred as a result of a relocation request.

**Sys. Programmer** A logic error occurred in the &mon dispatcher processing. Report the error to your technical support representative.

**DIS0012 Task group \$1 added tasks \$2 priority \$3**

**Explanation** The task group \$1 was successfully added with \$2 tasks and a priority of \$3.

**DIS0013 Task group \$1 modified tasks \$2 priority \$3**

**Explanation** The task group \$1 was modified and now has \$2 tasks and a priority of \$3.

**DIS0014 Task group \$1 delete requested, waiting for \$2 users**

**Explanation** A request was issued to delete the task group \$1. Before it can be deleted, the task group must first be quiesced; that is, any users currently using the task group must terminate. When this message was issued, there were \$2 user(s) using the task group.

**Comp. Operator** This message generally occurs during termination of the SMARTS server environment. If the message continually repeats with the same number of users, it indicates that some users may have not terminated correctly and thus did not remove their use count from the task group. In this case, you may bypass the problem with a forced EOJ or, if diagnostic information is required, you may cancel the SMARTS server environment with a dump.

**DIS0015 Task group \$1 deleted successfully**

**Explanation** Task group \$1 successfully finished quiescing and all resources acquired for the task group have been freed.

**DIS0016 Task group \$1 add failed rc=\$2 fdbk=\$3**

**Explanation** An attempt to add task group \$1 failed with return code \$2 and feedback code \$3.

**System Action** The task group is not added.

RC	FB	Reason
08	20	Changes already in progress; retry the operation
08	32	Insufficient storage for control blocks (in gen. buffer pool)
08	52	Operating system attach failed (maximum tasks attached or OS attach failure)

If any other combination appears, report the error to your technical support representative.

**Sys. Programmer** The task group add can fail due to a lack of resources on the system. The following lists return code/feedback code combinations that may legitimately occur and their causes:

RC	FB	Reason
08	20	Changes already in progress; retry the operation
08	32	Insufficient storage for control blocks (in gen. buffer pool)
08	52	Operating system attach failed (maximum tasks attached or OS attach failure)

If any other combination appears, report the error to your technical support representative.

**DIS0017 Task group \$1 modify failed, rc=\$2 fdbk=\$3**

**Explanation** An attempt to modify task group \$1 failed with return code \$2 and feedback code \$3.

**System Action** Depending on the point where the problem occurred, some of the modifications may have been implemented. Check the task group status with the UCTRL utility.

RC	FB	Reason
08	20	Changes already in progress; retry the operation
08	32	Insufficient storage for control blocks (in gen. buffer pool)
08	52	Operating system attach failed (maximum tasks attached or OS attach failure)

If any other combination occurs, report the error to your technical support representative.

**Sys. Programmer** The task group modify can fail due to a lack of resources on the system. The following lists return code/feedback code combinations that may legitimately occur and their causes:

RC	FB	Reason
08	20	Changes already in progress; retry the operation
08	32	Insufficient storage for control blocks (in gen. buffer pool)
08	52	Operating system attach failed (maximum tasks attached or OS attach failure)

If any other combination occurs, report the error to your technical support representative.

**DIS0018 Get for wait buffer failed uid=\$1 tid=\$2 LU=\$3 Pgm=\$4**

**Explanation** During the dispatching cycle, an enlarged wait buffer was required for a task. An attempt to acquire this buffer failed. The failure occurred for userid \$1 running on LUname \$4 (TID \$2) for program \$4.

**System Action** The wait request must be ignored; therefore, the program hangs until the SMARTS server environment is brought down.

**Sys. Programmer** As this failure causes users to hang indefinitely, the general buffer pool allocations must be reviewed to ensure that the required buffers are available. Alternately, additional tasks can be allocated in the task group to avoid long wait lists building up on individual tasks.

**DIS0019 Attach failed rc=\$1 fdbk=\$2**

**Explanation** An attempt to attach a task failed with return code \$1 feedback code \$2.

**System Action** The request for which the attach was issued fails.

RC	FB	Reason
08	04	The IDENTIFY request failed prior to the attach (OS/390 MVS only)
08	08	The operating system ATTACH request failed
08	12	Add processing for the task termination ECB failed
08	16	The request failed because the maximum tasks were attached

**Sys. Programmer** Determine from the return and feedback codes why the attach failed and correct the problem. The following are the possible return and feedback codes and their cause:

RC	FB	Reason
08	04	The IDENTIFY request failed prior to the attach (OS/390 MVS only)
08	08	The operating system ATTACH request failed
08	12	Add processing for the task termination ECB failed
08	16	The request failed because the maximum tasks were attached

**DIS0020 Queue initialization completed successfully**

**Explanation** The SMARTS server environment builds a queue registration area in which all TIB queues in the system are registered at initialization. Also, the common TIB queues such as output, message, completion are built. This message indicates that this processing has finished successfully.

**DIS0021 Task group \$1 \$2 failed rc=\$3 fdbk=\$4**

**Explanation** A request \$2 to task group \$1 failed due to return code \$3 and feedback code \$4.

**System Action** The request is not processed.

RC	FB	Reason
08	20	Changes already in progress; retry the operation
08	32	Insufficient storage for control blocks (in gen. buffer pool)
08	52	Operating system attach failed (maximum tasks attached or OS attach failure)

If any other combination appears, report the error to your technical support representative.

**Sys. Programmer** The request may fail if insufficient system resources are available. The following lists return code/feedback code combination that may legitimately occur and their causes:

RC	FB	Reason
08	20	Changes already in progress; retry the operation
08	32	Insufficient storage for control blocks (in gen. buffer pool)
08	52	Operating system attach failed (maximum tasks attached or OS attach failure)

If any other combination appears, report the error to your technical support representative.

**DIS0022 Thread group \$1 \$2 failed rc=\$3 fdbk=\$4**

**Explanation** A request \$2 to thread group \$1 failed due to return code \$3 and feedback code \$4.

**Sys. Programmer** This may occur if the thread group parameters specified are invalid, or if insufficient resources are available to create/alter the thread group. The following lists the return code/feedback code combinations that may legitimately occur and their cause:

RC	FB	Reason
08	20	Changes already in progress; retry the operation
08	32	Insufficient storage for control blocks (in gen. buffer pool)
08	40	Insufficient storage for threads (in region/partition)
12	36	Invalid parameter value

If any other combination appears, report the error to your technical support representative.

**DIS0023 Waiting for task group \$1 to quiesce**

**Explanation** The system is waiting for task group \$1 to quiesce. Users are still using the task group.

**System Action** The system continues to wait on the task group and issue the message until such time as the users using the task group stop using it.

**Sys. Programmer** If the system fails to come down and continually issues this message, it indicates that there may be a problem with programs not removing their use count from a task group when they terminate. Report the error to your technical support representative and take a dump for diagnosis. If a dump of the situation is already available, you may terminate the SMARTS server environment using a forced EOJ.

DIS0024 Getmain for wait list failed

## INI — Initialization : Main Processing

---

**INI0001 SMARTS \$1 is initializing under \$2**

**Explanation** During initialization processing, the SMARTS server environment has received control from the operating system and is proceeding with normal initialization. The SMARTS server environment version is displayed in the message along with the operating system under which the SMARTS server environment is running.

**System Action** The SMARTS server environment continues with normal initialization processing.

**Comp. Operator** This is an informational message. No action is necessary.

**INI0002 Attach for \$1 subtask failed, return code \$2**

**Explanation** The SMARTS server environment attempted to attach a task for the indicated subtask; however, the attach failed with the printed return code.

**System Action** The initialization of the SMARTS server environment is aborted.

**Sys. Programmer** Determine why the attach failed. The printed return code was received from the ATTACH macro indicating that it had failed.

**INI0003 Initialization aborted**

**Explanation** A serious error identified in a previous message occurred during initialization of the SMARTS server environment.

**System Action** Initialization of the SMARTS server environment discontinues, terminating normally.

**Sys. Programmer** The error condition responsible for initialization failure was identified by a previous message on the operator's console. Correct the error condition and restart the SMARTS server environment.

**Comp. Operator** Report this error condition to the SMARTS server environment system programmer.

**INI0004 ESTAE processing failed for SMARTS subtask**

**Explanation** During subtask startup, the SMARTS server environment failed in its attempt to establish a recovery environment for the subtask.

**System Action** The subtask terminates with a bad return code; however, the results for the SMARTS server environment are unpredictable due to the loss of the subtask.

**Sys. Programmer** Determine why the environment could not be established. In an MVS environment, ESTAE processing has failed. Check the possible reasons why this can happen in the IBM manuals.

**Comp. Operator** Inform your systems programmer immediately and terminate the SMARTS server environment to avoid unpredictable results.

**INI0005      Nucleus size: \$1K below, \$2K above**

**Explanation** This message provides the current size of the nucleus, which includes the SMARTS server environment linked nucleus and all modules loaded to build the nucleus at startup including user exits.

The second figure is present only on a system where modules can reside above the line and shows the size of the modules, in bytes, loaded above the line. In this case, the sum of the two figures is the total size of the SMARTS server environment nucleus.

**INI0006            INIT-PGM \$1 abended, reply 'Y' to continue initialization**

**Explanation** The program specified in the message was specified as INIT-PGM in the SMARTS server environment sysparms. It was called on that basis and abended. The SMARTS server environment has recovered from the abend and is giving the operator the chance to continue initialization without the INIT-PGM having run successfully.

**System Action** The SMARTS server environment waits for a reply to the message. If the reply to the message is any character other than 'Y', initialization of the SMARTS server environment is aborted. When the reply is 'Y', initialization of the SMARTS server environment continues with the next INIT-PGM specification, if any.

**Sys. Programmer** Determine the reason for the abend of the program from the dump that was produced and correct the error. Otherwise, the specification should be removed to avoid abends with each initialization of the SMARTS server environment .

**Comp. Operator** Depending on whether the system can run successfully without running the particular INIT-PGM, reply 'Y' or 'N' and inform the systems programmer.

**INI0007            Patch character '\$1' already in use**

**Explanation** The patch character specified in the sysparms and displayed in this message is in use by another SMARTS server environment on the same system.

**System Action** Initialization of the SMARTS server environment is aborted.

**Sys. Programmer** The SMARTS server environment 'PATCHAR' sysparm must be unique for each SMARTS server environment in the system as this is used to uniquely identify the various instances of SMARTS server environments. Either choose another patch character or determine which other SMARTS server environment instance is using this patch character and why.

**INI0008      Nucleus load \$1**

**Explanation** Depending on various system conditions, the SMARTS server environment nucleus load may take some time. To insure that this is obvious, a message is issued about every six seconds. The first tells you that the load is 'STARTING', while the second and subsequent messages indicate that the load is 'IN PROGRESS'. This can and often does appear more than once and is, therefore, not an error but simply an information message.

**INI0009**      **\$1 error loading module \$2, reason code X'\$3'**

**Explanation**      The operation indicated in the message failed for the named module due to the displayed reason code. The following are the reason codes, depending on the operation.

Operation	Reason Code
BLDL	R0 contents after failed BLDL
LOAD	Abend code with which the load would have abended if the program hadn't specified an error exit

**System Action**      Depending on the need for the named module, the SMARTS server environment continues processing.

**Sys. Programmer**      Determine why this module is being loaded and if there is a valid reason that it should be loaded; determine why the load is failing.

**INI0010**      **Module not found: '\$1'**

**Explanation**      The main initialization routine of the SMARTS server environment could not locate the module indicated. \$1 is the module name.

**System Action**      Initialization of the SMARTS server environment discontinues.

**Sys. Programmer**      This is a serious error and may indicate corruption of the SMARTS server environment load library. Examine the library to ensure that the specified module is present. It may be necessary to restore that module or the entire library. This error usually means that a module has been inadvertently deleted from the library and does not, in general, indicate any physical error.

**Comp. Operator**      Notify the SMARTS server environment system programmer of this error.

**INI0012**      **Nucleus '\$1' invalid version/release '\$2'; expected**

**Explanation**      The nucleus named in the message is not at the version/ release level required by the initialization module used.

**System Action**      Initialization processing of the SMARTS server environment terminates.

**Sys. Programmer**      The SMARTS server environment nucleus \$1 loaded at initialization by the initialization module does not reflect the version/release level \$3 at which the initialization module was assembled, indicating a mismatch. Check that the module you think you are loading is really the module that you are loading. The version/release level \$2 of the nucleus you have loaded should help you to identify which level of the SMARTS server environment nucleus you have loaded, while \$3 tells you which level the initialization module is at.

**INI0013 Module loaded: '\$1'****INI0014 BP Create for \$1 Buffer Pool returned status='\$2'**

**Explanation** The SMARTS server environment attempted to generate the \$1 buffer pool. The create request returned a status of \$2. \$2 is the hex representation of the status returned from the buffer pool management routines. The first two bytes are the return code from the request and the second two bytes are the reason code. See appendix H, Request Status, starting on page for detailed information.

**System Action** When the return code is "4", processing continues. When the return code is greater than "4", the buffer pool has not been built. If the buffer pool is the 'workpool' buffer pool, this is required for SMARTS server environment execution and therefore initialization fails. For any other buffer pool, initialization continues; however, the buffer pool is not available for use.

**Note:** When the return code is "8" and the reason "24" is indicated, this error might be caused by starting the EntireX Broker twice. To do so might lead to a shortage of SYSDEF dataspace that can be a potential reason this message.

**Sys. Programmer** If the return code is greater than "4", determine why the buffer pool creation failed. Generally this happens only due to a shortage of storage. When the return code is "4", report the circumstances of the message to your technical support representative.

**INI0015 Warning: Module \$1 required to be reentrant is NOT**

**Explanation** Various SMARTS server environment nucleus routines or nucleus user exit routines need to be coded reentrant and linked with the RENT option in order to guarantee integrity of the SMARTS server environment.

Module \$1 was found *not* to be reentrant and therefore may result in loss of integrity or nucleus abends.

**System Action** Initialization continues.

**Sys. Programmer** Ensure module \$1 is reentrant.

**INI0016 Add failed for \$1 resource, rc=\$2 fdbk=\$3**

**Explanation** During initialization, the SMARTS server environment adds various resources to the general resource pool for use at a later stage. In this case, the add for resource \$1 failed with return code \$2 and feedback code \$3. See appendix H, Request Status, starting on page for detailed information.

**System Action** Initialization terminates.

**Sys. Programmer** This indicates a logic error in the initialization of the SMARTS server environment. Report the contents of the message and the circumstances that cause it to appear to your technical support representative.

**INI0017**            **BPDelete for \$1 BP returned status=x'\$2'**  
**Explanation**        While attempting to delete the buffer pool \$1, a status of \$2 was returned.  
**System Action**      Processing continues, however, storage may be lost if the delete request actually failed. Failure is indicated in the first halfword of the status \$2. When this is x'0008' or greater, the delete request failed. See appendix H, Request Status , starting on page for detailed information.  
**Sys. Programmer**    An unexpected situation has occurred. \$2 represents the return code and feedback code from the BPDELETE request. Report this, along with any other messages at the time of the error, to your technical support representative.

**INI0018**            **Logic error in module \$1 at offset \$2**  
**Explanation**        The SMARTS server environment encountered a logic error in module \$1 at offset \$2.  
**Sys. Programmer**    Report the error and the circumstances surrounding the error to your technical support representative.

**INI0019**    **CRSVATBL not loaded into SVA**

**INI0020**    **System adapter (COMSIP) not initialized**

**INI0021**    **Return code from PRODID AUTH not zero - RC=\$1**

**INI0022**    **PRODID-DEFINE - RC = \$1/\$2**

**INI0023**    **Not enough real storage for initialization**

## **INP — Initialization : Parameter Processing**

---

**INP0003**            **Invalid data for Keyword \$1, '\$2'**  
**Explanation**        While processing the SMARTS server environment initialization parameters specified in the SYSPARM file or the PARM field text, a value was given for the specified keyword that was not recognized as valid.  
**System Action**      The SMARTS server environment terminates with a return code of 8.  
**Sys. Programmer**    Locate the source of the error, correct it, and resubmit the job.  
**Comp. Operator**    Notify the SMARTS server environment system programmer.

**INP0007**      **Not enough storage to process SYSPARMs**  
**Explanation**      While processing SMARTS server environment initialization parameters, an insufficient amount of main storage was available to process the SYSPARM file or the PARM field text.  
**System Action**      The SMARTS server environment terminates with a return code of 8.  
**Sys. Programmer**      Increase the region or partition size by 2K and resubmit the job.  
**Comp. Operator**      Notify the SMARTS server environment system programmer.

## LOD — Program Management Services

---

**LOD0001**      **Invalid address in program linkage function**  
**Explanation**      Associated with OS LOAD, LINK, XCTL, or DELETE SVCs and the SMARTS server environment's COLOAD, COLINK, COXCTL, or CODEL functions, this message indicates that the parameter list passed contains invalid addresses.  
**System Action**      The application program is abnormally terminated and a SMARTS server environment online dump is taken.  
  
Register 1 contains the address of the parameter list.  
The first address in the parameter list should be the address of the program name being loaded.  
If an OS SVC was used, the second address should be 80000000 (hex).  
If a SMARTS server environment function was used, only the first address is checked for validity.  
**Term. Operator**      An error occurred in the application program. Notify the person responsible for such errors.  
  
Register 1 contains the address of the parameter list.  
The first address in the parameter list should be the address of the program name being loaded.  
If an OS SVC was used, the second address should be 80000000 (hex).  
If a SMARTS server environment function was used, only the first address is checked for validity.  
**Appl. Programmer**      The following may help to determine how the error occurred:  
Register 1 contains the address of the parameter list.  
The first address in the parameter list should be the address of the program name being loaded.  
If an OS SVC was used, the second address should be 80000000 (hex).  
If a SMARTS server environment function was used, only the first address is checked for validity.

<b>LOD0002</b>	<b>No storage for XCTL parameter list copy</b>
<b>Explanation</b>	In order to effect a XCTL/COXCTL request, the SMARTS server environment attempts to acquire storage to make a copy of the parameter list being passed. The request to acquire this storage failed.
<b>System Action</b>	The application program is abnormally terminated and a SMARTS server environment online dump is taken.  Register 1 contains the address of the parameter list. The first address in the parameter list should be the address of the program name being loaded. The second address in the parameter list should be 80000000 (hex).
<b>Sys. Programmer</b>	The storage referred to in this message is acquired from the general buffer pool. Review the storage estimates for this region to ensure that such as shortage does not occur again.  Register 1 contains the address of the parameter list. The first address in the parameter list should be the address of the program name being loaded. The second address in the parameter list should be 80000000 (hex).
<b>Appl. Programmer</b>	The following may help to determine how the error occurred:  Register 1 contains the address of the parameter list. The first address in the parameter list should be the address of the program name being loaded. The second address in the parameter list should be 80000000 (hex).
<b>LOD0003</b>	<b>Invalid resume PSW at exit from LINKed or XCTLed</b>
<b>Explanation</b>	The return from the application that was passed control using OS LINK, XCTL, or the SMARTS server environment's COLINK or COXCTL functions indicates that the application program modified the storage in the thread used to save the status of the calling program.
<b>System Action</b>	The application program is abnormally terminated and a SMARTS server environment online dump is taken.
<b>Term. Operator</b>	An error occurred in the application program. Notify the person responsible for such error.
<b>LOD0004</b>	<b>Program \$1 LOAD for \$2 disallowed by exit ULSRPSFS</b>
<b>Explanation</b>	During an attempt by the application program to load a module using OS LOAD, LINK, XCTL, or the SMARTS server environment's COLOAD, COLINK, or COXCTL functions, the installation user exit ULSRPSFS chose not to allow the application to load the requested program.
<b>System Action</b>	The application program is abnormally terminated and a SMARTS server environment online dump is taken.
<b>Term. Operator</b>	Contact the application programmer.

## OPC — Operator Communication

---

**OPC0000      SMARTS is initialized**

**Explanation**      The SMARTS server environment initialized successfully and is now ready to accept users and operator commands.

**Comp. Operator** This is an informational message only; no action is required.

**OPC0001      \$1 \$2 completed**

**Explanation**      The SMARTS server environment computer operator command \$1 with operand \$2 was accepted and processed. This message is sent to the console at the completion of processing for the following commands: ADD, DELETE, IGNORE, RESTART, CANCEL, STALL, UNSTALL, SETCTL, UNCTL, and LOGOFF.

**System Action**      The appropriate action is performed according to the command entered.

**Comp. Operator** This is an informational message only; no action is required.

**OPC0002      \$1: \$2 is not logged on**

**Explanation**      The user ID \$2 referred to in the SMARTS server environment computer operator command \$1 is not logged on to SMARTS.

**System Action**      The command is ignored.

**Comp. Operator** The operand field (\$2) must be a valid user ID for a user who is currently logged on to the SMARTS server environment.

**OPC0003      Unrecognized command: \$1**

**Explanation**      The SMARTS server environment operator (or a privileged terminal user) entered a SMARTS server environment system command, but the SMARTS server environment was unable to identify the command verb or not enough characters were entered to uniquely identify the verb.

**System Action**      The command is not executed.

**Comp. Operator** This message is sent to the console that entered the erroneous data (or the master console, if entered by a terminal user). Examine the failing command and reenter using a valid verb.

- OPC0004**      **\$1 \$2 \$3ID=\$4 - Device not supported**
- Explanation**      This message is associated with SMARTS server environment computer operator commands. Device support for TID \$3 is not included in the SMARTS server environment nucleus, or TID \$3 is either a batch or attached TID.
- System Action**      The command is ignored unless \$3 is a member of group \$2. In that case, the remainder of group \$2 is processed.
- Comp. Operator**      After verifying that the correct TID or LID was entered in the command, notify the system programmer, who should include the device support modules for TID \$3 in the next link of the SMARTS server environment nucleus.
- 
- OPC0005**      **Invalid EOJ command format**
- Explanation**      This message is associated with SMARTS server environment computer operator commands. It indicates that the EOJ command was invalid.
- System Action**      The command was ignored.
- Comp. Operator**      Refer to the explanation of the use of the EOJ command.
- 
- OPC0006**      **\$1 \$2 - TID or group invalid or not defined**
- Explanation**      This message is associated with SMARTS server environment computer operator commands. In command \$1, \$2 is invalid. Here, \$2 must be a valid TID, LID, user ID, group name, or ALL.
- System Action**      The command is ignored.
- Comp. Operator**      Reenter the command with a valid \$2 field.
- 
- OPC0007**      **\$1 \$2 - already active**
- Explanation**      This message is associated with the SMARTS server environment computer operator ADD and RESTART commands. Terminal (online) \$2 was activated by a SMARTS server environment ADD command at initialization or by a previous ADD or RESTART command and has not been IGNORED.
- System Action**      The command is ignored.
- Comp. Operator**      The terminal is already in the active state. A terminal must be in the nonactive state (via the IGNORE command) before a RESTART command can be issued for it.
- 
- OPC0010**      **\$1 \$2 multiple terminal cancel not allowed**
- Explanation**      This message is associated with the SMARTS server environment computer operator CANCEL command. It indicates that more than one TID, TIBNAME, or user ID (\$2) was entered for command \$1.
- System Action**      The command is ignored.
- Comp. Operator**      Only one terminal may be cancelled at a time. Here, \$2 may only be one valid TID, TIBNAME, or user ID.

**OPC0011      \$1 \$2 - Can't do it now, APPC TID is in RECEIVE state**

**Explanation** This message is issued if an attempt is made to CANCEL or LOGOFF an APPC TIB while this TIB is waiting for input.

**System Action** The command is ignored.

**OPC0012      \$1 \$2 - CANCEL delayed, in SMARTS or PV program**

**Explanation** This message is issued in response to a CANCEL command entered by the computer operator. It indicates that a privileged program (UQ, UEDIT, etc.) or a SMARTS server environment service routine had control of terminal \$2 when the CANCEL command (\$1) was issued.

**System Action** The program running on terminal \$2 has been flagged to be terminated at the end of the current process.

**Comp. Operator** Software AG does not recommend cancelling privileged programs or the service routine; however, by reentering the CANCEL command, the computer operator may be able to cancel the current process.

**OPC0015      \$1 \$2 - Terminal has already been ignored**

**Explanation** An IGNORE command was issued for a terminal (\$2) that had already been ignored.

**System Action** The command is ignored.

**Comp. Operator** The terminal or line is already ignored; no further action need be taken.

**OPC0017      \$1 \$2 \$3ID=\$4 rolled out - CANCEL before IGNORE**

**Explanation** TID \$3 is rolled out and cannot be ignored.

**System Action** The command is ignored unless \$3 is a member group, in which case processing of the group continues.

**Comp. Operator** The program that is active for the terminals (\$3) being ignored must be allowed to finish processing or be cancelled before the terminal can be ignored.

**OPC0018      \$1 \$2 \$3ID=\$4 not ignored - active in thread**

**Explanation** The program associated with the terminal is active in the thread and must be cancelled before the terminal can be ignored.

**System Action** The command (\$1) is ignored unless \$3 is a member of group \$2, in which case processing of the group continues.

**Comp. Operator** If the program associated with a terminal remains active in the thread, the proper action is to cancel it rather than to ignore the terminal. This condition can be the result of a missing I/O interrupt or an outstanding console reply. Check for these conditions before cancelling a program.

**OPC0020 Error on destination code \$1**

**Explanation** This message is associated with SMARTS server environment computer operator commands. The SMARTS server environment was unable to issue its hello message to all the terminals in the SMARTS server environment for one or more of the following reasons:

•	A security violation occurred because TID 1 was not authorized to send a class 2 message.
•	DEST CODE \$1 was translated by the SMARTS server environment to define more than 100 terminals.
•	DEST CODE \$1 was higher than the highest code defined to the SMARTS server environment.
•	DEST CODE \$1 was not defined in TIBTAB.
•	DEST CODE \$1 defined a terminal that was not authorized to receive class 2 messages.

**System Action** The destination code indicated by DEST CODE \$1 is bypassed; that is, the SMARTS server environment hello message is not sent to the terminals defined by DEST CODE \$1.

**OPC0021 \$1 \$2 already active**

**Explanation** This message is associated with the SMARTS operator commands ADD and RESTART:

•	For an ADD command, the terminal (\$2) cannot be ADDED because it is already active to another user or job within the system.
•	For a RESTART command, the terminal (\$2) is already active under SMARTS.

**System Action** The command is ignored.

**Comp. Operator** For an ADD command, the terminal must first be released from its current user/owner. For a RESTART command, the terminal must be deactivated using an IGNORE operator command before issuing the RESTART command.

**OPC0022 \$1 \$2 \$3ID=\$4 is enqueued and cannot be restarted**

**Explanation** TID \$3 is in a SMARTS queue and cannot be restarted.

**System Action** The command \$1 is ignored unless \$3 is a member of a group (\$2), in which case processing of the remainder of the group continues.

**Comp. Operator** This message indicates a logic problem in the SMARTS nucleus. Contact your technical support representative.

**OPC0023      Job '\$1' is using SMARTS functions, enter 'Y' to Continue**

**Explanation**      The SMARTS server environment was unable to perform the EOJ operation specified by the computer operator because the batch job indicated by \$1 was using a SMARTS server environment function at the time the EOJ command was issued.

**System Action**    The EOJ command is ignored.

**Comp. Operator**    Wait until the batch job \$1 terminates and then reissue the EOJ command.

**OPC0024      \$1 \$2 - ENQ SVC routine in control - reenter command**

**Explanation**      The SMARTS server environment was unable to perform the CANCEL operation requested by the computer operator because the TID specified by the computer operator is active in the thread and is currently waiting for an ENQ/LOCK function to complete.

**System Action**    The CANCEL command is ignored.

If the cause of the wait cannot be determined or the resource cannot be made available, issuing a second CANCEL command for this TID cancels the program associated with this TID, leaving the resource for which the program was waiting enqueued with no associated program to later dequeue it.

The SMARTS server environment program termination cleanup is unable to DEQUEUE/UNLOCK this resource since it is in the waiting status.

Use the SMARTS server environment utility UENQ to DEQUE/ UNLOCK the resource left when the TID was cancelled; however, ensure that you do not DEQUEUE/UNLOCK the wrong resource.

**Sys. Programmer**    Determine why the program associated with this TID is waiting to ENQ/LOCK on a resource and attempt to make the resource available.

If the cause of the wait cannot be determined or the resource cannot be made available, issuing a second CANCEL command for this TID cancels the program associated with this TID, leaving the resource for which the program was waiting enqueued with no associated program to later dequeue it.

The SMARTS server environment program termination cleanup is unable to DEQUEUE/UNLOCK this resource since it is in the waiting status.

Use the SMARTS server environment utility UENQ to DEQUE/ UNLOCK the resource left when the TID was cancelled; however, ensure that you do not DEQUEUE/UNLOCK the wrong resource.

**Comp. Operator**    Before issuing the CANCEL command for this TID a second time, seek the assistance of the SMARTS server environment system programmer.

**OPC0025 \$1 command invalid for VTAM / ACCESS****OPC0026 DYNALLOC issued for SYSPRINT**

**Explanation** While attempting to print something to SYSPRINT, the OC task found that the DD/DLBL statement did not exist in the job control. Thus, the DD/DLBL statement was dynamically allocated to make it available for use.

**Sys. Programmer** To avoid this message, the SYSPRINT DD/DLBL statement must be coded in the SMARTS server environment job control.

**OPC0027 \$1 command operand must be either 'START' OR 'STOP'**

**Explanation** This message is associated with SMARTS server environment VTAM support. An invalid operand field was entered with the VTAM command.

**System Action** The command is ignored.

**Comp. Operator** The operand field for the VTAM command must be STOP or START.

**OPC0028 Error during DYNALLOC Error Code X'\$1', Info Code '\$2'**

**Explanation** The SMARTS server environment OC task attempted to print to the SYSPRINT DD/DLBL but found that it did not exist in the job control. An attempt to allocate it dynamically failed. For MVS systems, \$1 and \$2 are the error and information codes, respectively, returned by the DYNALLOC function.

**System Action** The attempt to print is terminated.

**Sys. Programmer** Using the diagnostic information, determine why the dynamic allocation of the DD/DLBL statement failed. If the problem is related to the installation, correct the error. If not, provide the details of the error to your technical support representative.

**OPC0029 \$1 trace entries lost due to insufficient buffer space**

**Explanation** Tracing is active in the system where a certain amount of storage has been reserved for the trace areas. As the buffer areas fill, they may be written to CAPTURE and reinitialized or simply reinitialized if the data is not being CAPTUREd.

When a buffer is full, an attempt is made to find the next 'reinitialized' or available trace buffer. If this cannot be found due to delays in CAPTURE or the speed of the machine, the requested data cannot be traced and therefore trace entries are lost. The message registers the number of trace entries \$1 lost in the previous 60 seconds or so.

**System Action** The trace entry is lost and processing continues.

It may occur if a minimum of trace buffers is made available and large numbers of trace records are being written. This can occur when trace data is being CAPTUREd and the volume of tracing is filling the allocated buffers faster than they can be written to CAPTURE.

In this case, allocate enough buffers to ensure that the filled buffers are CAPTUREd and reinitialized *before* the 'free' buffers are filled again.

- Sys. Programmer** It is unlikely that this message will be seen when trace data is not being CAPTUREd as the CAPTURE buffers are reinitialized for use as soon as they are filled.
- It may occur if a minimum of trace buffers is made available and large numbers of trace records are being written. This can occur when trace data is being CAPTUREd and the volume of tracing is filling the allocated buffers faster than they can be written to CAPTURE.
- In this case, allocate enough buffers to ensure that the filled buffers are CAPTUREd and reinitialized *before* the `free' buffers are filled again.
- OPC0030 SMARTS termination in progress**
- Explanation** This message is associated with the SMARTS server environment operator communications processor and is the normal response to an EOJ (shutdown) request from the operator or a privileged terminal user.
- System Action** The SMARTS server environment begins its normal termination processing. Thread activity is quiesced, no user is allowed to enter any data, and active programs are terminated. No further operator communications are accepted.
- Comp. Operator** This message is for information only; no action is required.
- OPC0031 SMARTS terminated**
- Explanation** This message is associated with the SMARTS server environment operator communications processor and is the normal response to an EOJ (shutdown) request from the operator or a privileged terminal user. The SMARTS server environment finished its normal termination processing and is exiting the operating system.
- System Action** The SMARTS server environment terminates normally with return code of zero.
- Comp. Operator** This message is for information only; no action is required.
- OPC0032 LOGON now disallowed**
- Explanation** The DISALLOW LOGON command entered by the computer operator completed successfully.
- System Action** Future LOGON requests are not honored; however, all users who are already logged on to the SMARTS server environment are able to continue until they log off.
- Comp. Operator** Issue the ALLOW command when ready to allow new users to log on to the SMARTS server environment. Refer to the information about this command.
- OPC0033 LOGON now allowed**
- Explanation** The ALLOW LOGON command entered by the computer operator completed successfully.
- System Action** The SMARTS server environment resumes honoring LOGON requests from terminals.
- Comp. Operator** For details, refer to the information about this command.

- OPC0034**      **RJE now disallowed**  
**Explanation**    The DISALLOW RJE command entered by the computer operator completed successfully.  
**System Action**    Future RJE requests from application programs receive a return code indicating that RJE is disallowed.  
**Comp. Operator**    Issue the ALLOW command when ready to allow RJE requests from application programs. For details, refer to the information about this command.
- OPC0035**      **RJE now allowed**  
**Explanation**    The ALLOLW RJE command entered by the computer operator completed successfully.  
**System Action**    The SMARTS server environment resumes honoring RJE requests from application programs.  
**Comp. Operator**    Refer to the information about the ALLOW command.
- OPC0036**      **LOADs disallowed. DEQ-ed: \$1**  
**Explanation**    The DISALLOW ULIB command entered by the computer operator completed successfully.  
**System Action**    All future use of the ULIB utility results in an error message indicating that ULIB is disallowed.  
**Comp. Operator**    Issue the ALLOW command when ready to allow use of the ULIB utility. Refer to the information about this command.
- OPC0037**      **LOADs allowed. ENQ-ed: \$1**  
**Explanation**    The ALLOW ULIB command entered by the computer operator completed successfully.  
**System Action**    The ULIB utility resumes honoring requests.  
**Comp. Operator**    Refer to the information about the ALLOW command.
- OPC0038**      **UQDEFAULT is now**  
**Explanation**    The DISALLOW UQ command entered by the computer operator completed successfully.  
**System Action**    The UQ utility does not allow the H, R, S, C, DE, and OC commands to be issued against any batch job that does not have the special UQ JCL security statements (/\*UQ USER ID, /\*UQ ACCOUNT, etc.).
- Specifying UQDEFAULT=NO in the initialization parameters for the SMARTS server environment accomplishes the same thing.

**OPC0039 UQDEFAULT is now `ALLOW`**

**Explanation** The ALLOW UQ command entered by the computer operator completed successfully.

**System Action** The UQ utility does not secure the H, R, S, C, DE, or OC commands when they are issued against a batch job that does not have the special UQ JCL security statements (/\*UQ USER ID, /\*UQ ACCOUNT, etc.).

Specifying UQDEFAULT=YES in the initialization parameters for the SMARTS server environment accomplishes the same thing.

**OPC0040 Invalid data for ALLOW/DISALLOW command**

**Explanation** The ALLOW or DISALLOW command entered by the computer operator indicates that the parameter supplied was not one of following: LOGON, RJE, ULIB, or UQ. For details, refer to the information about the ALLOW and DISALLOW commands.

**System Action** The command is ignored.

**OPC0041 \$1 \$2 scheduled via terminal ENQ**

**Explanation** The operator issued a \$1 command and the request was processed successfully by ENQing the user's terminal.

**System Action** The \$1 command is processed asynchronously to eventual completion.

**Term. Operator** The user is logged off asynchronously.

**Comp. Operator** The \$1 command has been processed successfully.

**OPC0043 IGNORE accepted**

**Explanation** This message, associated with the SMARTS server environment operator communications processor, is the normal response to an IGNORE operator command. The system operator or a privileged terminal user entered a request to ignore input/output requests for a terminal, line, or group of terminals and lines. The IGNORE command was successfully processed.

**System Action** Normal processing continues. No data is entered from or sent to the designated terminal(s)/line(s).

**Comp. Operator** This message is sent to the console that entered the command or to the master console if entered by a terminal user. The requested terminal(s) or line(s) are now allocated to the SMARTS server environment and are ineligible for allocation by other teleprocessing systems or jobs. It is necessary to issue a DELETE operator command to allow the terminal(s) or line(s) to be allocated by another job.

---

<b>OPC0045</b>	<b>Module '\$1' not found for termination</b>
<b>Explanation</b>	During termination processing, the SMARTS server environment attempted to load module \$1; however, the module was not found in the load libraries available to the SMARTS server environment.
<b>System Action</b>	If the module is not required, processing continues as normal. If the module is required for termination, a message is issued indicating that termination may not finish successfully without this module.
<b>Comp. Operator</b>	Determine if the module is necessary for correct termination and if so make it available in the load libraries available to the SMARTS server environment.
<b>OPC0046</b>	<b>I/O error for module '\$1'</b>
<b>Explanation</b>	During termination processing, the SMARTS server environment attempted to load module \$1. During load processing, an error occurred and the module could not be loaded.
<b>System Action</b>	If the module is not required, processing continues as normal; however, if it is required, a warning is issued that termination may not complete correctly without this module.
<b>Sys. Programmer</b>	Usually, the operating system issues a message in relation to the error. Use this information to determine why an error occurred loading the module and correct the error.
<b>OPC0048</b>	<b>Module \$1 \$2 error, ABEND code X'\$3'</b>
<b>Explanation</b>	During termination processing, the SMARTS server environment attempted to load module \$1. Either the BLDL or the LOAD failed as indicated by \$2. The error code \$3 is the ABEND code returned from the load if load processing failed. If BLDL processing failed, this code is the contents of R0 after the failed BLDL.
<b>System Action</b>	If the module is not required, processing continues as normal. If the module is required, a warning message is issued indicating that termination processing may not complete successfully.
<b>Sys. Programmer</b>	Based on the provided diagnostic information, determine why the function failed and correct the error for the next SMARTS server environment termination.
<b>OPC0049</b>	<b>Return code \$1 from module '\$2'</b>
<b>Explanation</b>	During startup or termination processing, module \$2 was called and returned a return code \$1.
<b>System Action</b>	If this happens during startup, then in most cases the initialization process is stopped. During termination, processing continues with the next module.
<b>Sys. Programmer</b>	If the module is a SMARTS server environment module, report the return code to your technical support representative as it indicates some form of unexpected error. If the module is a user module or exit, determine why the module returned the code it did and correct the problem or the module to cause a zero return code as required.

**OPC0050      Successful termination not possible without module**

**Explanation**      During termination processing, an error was encountered with a module that was required to successfully bring about normal termination. The module \$1 did not successfully execute and therefore termination may not be normal. The reason that the module did not successfully execute is clear from the message immediately preceding this related to the same module name.

**System Action**      Processing continues with the next termination module.

**Sys. Programmer**      Determine the problem with the module and correct it to insure normal termination of the SMARTS server environment.

**OPC0052      IGNORE rejected due to program active, CANCEL first**

**Explanation**      The SMARTS server environment was unable to honor the computer operator command IGNORE because the program running under the specified terminal was rolled out waiting for an event to complete (such as a call to ADABAS or a timed rollout).

**System Action**      The IGNORE command is not honored.

<ul style="list-style-type: none"> <li>• will not complete in a reasonable amount of time, such as a program that has rolled out for an extremely long time; or</li> </ul>
<ul style="list-style-type: none"> <li>• may never complete at all, such as an application program waiting for ADABAS when ADABAS has terminated.</li> </ul>

If this is the case, cancel the program (CANCEL command) before the terminal is ignored. Refer to information about the CANCEL command.

**Comp. Operator**      The continual appearance of this message normally indicates that the application program is waiting for an event that

<ul style="list-style-type: none"> <li>• will not complete in a reasonable amount of time, such as a program that has rolled out for an extremely long time; or</li> </ul>
<ul style="list-style-type: none"> <li>• may never complete at all, such as an application program waiting for ADABAS when ADABAS has terminated.</li> </ul>

If this is the case, cancel the program (CANCEL command) before the terminal is ignored. Refer to information about the CANCEL command.

**OPC0053      SMARTS threads quiesced**

**Explanation**      The SMARTS server environment termination processing has reached the 'WAIT WORK' status; that is, the point where it expects all threads to be quiesced.

**System Action**      Termination processing proceeds to the next stage.

**Comp. Operator**      If it seems that the threads will never go to wait work, either force 'normal' termination using the 'EOJ FORCE' command, or cancel the SMARTS server environment with a dump to provide some diagnostic information for the problem.



**Sys. Programmer** Determine the cause of the delayed cancel. The `must-complete' status is set while writing the capture file, RJE processing, or operator command processing. If the cause cannot be determined, cancel the SMARTS server environment command with a dump and contact your support representative.

**Comp. Operator** Wait for the delayed cancel to take effect. If the cancel has not completed after a reasonable time period, contact the system programmer at your installation.

**OPC0059 \$1 \$2 - command not applicable to this TIB**

**OPC0060 \$1 \$2 failed - device already deleted**

**Explanation** An attempt was made to delete a device (\$2) that was already deleted.

**Comp. Operator** No action is necessary as the terminal or line is already deleted.

**OPC0062 \$1 \$2 flagged but not scheduled: TIB inconsistent or in**

**Explanation** The command \$1 was issued against the user defined by \$2; however, the command could not complete successfully as the TIB for the user was in an invalid state or was waiting in one of the SMARTS server environment's internal queues. In this case, the requested operation can only be marked for the user.

**Sys. Programmer** Use UCTRL subfunctions US/QO/PL/SG/TL to determine if there is a lock situation somewhere causing the TIB to be hung in a queue.

If there is, try to remove the reason for the hang; for example, by cancelling the user locking up a thread or subtask.

If the TIB is not on any queue, this message generally indicates a logic error that should be reported to your technical support representative.

To get rid of the TID/User when it is not on a queue, use the `FORCE' command. If the same message occurs when the `FORCE' command is issued, the TIB is in a state that could cause an ABEND if the TIB was FORCED. In this case, the TIB and user ID are lost until the next time the SMARTS server environment is brought down and back up again.

**Comp. Operator** When a command fails in this way, contact the systems programmer.

**OPC0063 EOJ command rejected due to verification**

**Explanation** The systems programmer has specified an eight-byte character string which must be entered with the EOJ command to ensure that it comes from an authorized user knowing the verify data. This message indicates that a verify string must be entered and/or the incorrect verify string was provided.

**System Action** The EOJ command is ignored.

**Sys. Programmer** Ensure that the verify string is known to users authorized to bring the SMARTS server environment down.

**Comp. Operator** If you are authorized to bring the SMARTS server environment down, your systems programmer must provide you with the verify data in order for the EOJ command to work.

**OPC0064      REVIEW termination program attached**

**Explanation** When the Review product is installed in the system, for the SMARTS server environment, it is necessary for a Review EOJ program to be attached. This message indicates that this has just occurred.

**System Action** The EOJ command must be entered again after the Review EOJ program has terminated successfully.

**Comp. Operator** Wait for the successful completion of the Review EOJ program and issue the EOJ command again.

**OPC0065      CAPTURE functions not active**

**Explanation** An operator command related to CAPTURE processing was entered; however, CAPTURE is not active in the system.

**OPC0066      -> \$1**

**Explanation** This message is received in response to a DCTRL operator command. This causes a header line to be printed with this message number followed by a list of control TIDs.

**OPC0067      -> \$1**

**Explanation** This message is issued in response to a PLIST or TLIST operator command. The first message contains a header describing the format of the lines to follow and is followed by the lines of information requested.

**OPC0068      >>> FORCED <<< termination in progress**

**Explanation** The SMARTS server environment EOJ was requested with the FORCE option. The SMARTS server environment terminates without waiting for various termination processes to complete.

**OPC0069      Waiting for \$1 user(s) to logoff**

**Explanation** The SMARTS server environment EOJ is in progress and a logoff has been issued for all users. Having issued the logoff, the SMARTS server environment must wait until all users are logged off. The message is issued periodically with the number of users left to logoff (\$1).

**System Action** The SMARTS server environment waits for a number of seconds and rechecks to see if all users are logged off.

**Comp. Operator** In error situations, all users may not logoff. The operator can then FORCE each individual user or can reissue the EOJ command with the `FORCE' option to bypass the check.

**OPC0070 SMARTS detected mother task ABEND**

**Explanation** The SMARTS server environment main task ABENDED. Diagnostic ZAB messages are issued along with this message.

**OPC0071 SMARTS detected operator cancel**

**Explanation** The SMARTS server environment recovery routines detected an operating system cancel request.

**System Action** After certain necessary cleanup, the cancel is allowed to continue without any attempt at recovery.

**OPC0072 Attach failed for program \$1 return code \$2**

**Explanation** The OC tried to attach a program

•	as a result of the STARTUPPGM parameter;
•	as a result of an attach due to the contents of ULPGMTAB;
•	as a result of a `USER' operator command; or
•	at EOJ processing for Review.

The attach failed due to return code \$2. The reason for failure is as follows:

04	The requested program could not be found.
08	A security violation has occurred.
12	An invalid program name was provided.
16	A logic error has occurred while processing the request.
20	No eligible thread was available in which to run the program.
24	No TIB was available on which to run the program.
28	A valid and supported terminal type could not be established.
32	Insufficient storage in the general buffer pool for request.

**System Action** No action is taken, the program is simply not attached.

04	The program is not available to run in a thread. This may mean that the program is not in the loadlibs/CILs available to the SMARTS server environment or it may be the result of installation options.
08	Access to the program has been denied by a security exit.
12	A valid program name must start with an alpha character and be no more than 8 characters long.
16	Report this error to your technical support representative.

20	The program you requested is cataloged with a region size larger than any thread currently defined. Either recatalog the program to a size small enough to run in one of the currently defined threads or define a thread large enough to run the program.
24	Define more dynamic TIBs in your TIBTAB.
28	Device support for at least one of the 3270 family of terminals must be defined in your TIBTAB (for example, 3278).
32	Review your storage estimates for your general buffer pool.

**Sys. Programmer** Correct the error based on the return code as follows:

04	The program is not available to run in a thread. This may mean that the program is not in the loadlibs/CILs available to the SMARTS server environment or it may be the result of installation options.
08	Access to the program has been denied by a security exit.
12	A valid program name must start with an alpha character and be no more than 8 characters long.
16	Report this error to your technical support representative.
20	The program you requested is cataloged with a region size larger than any thread currently defined. Either recatalog the program to a size small enough to run in one of the currently defined threads or define a thread large enough to run the program.
24	Define more dynamic TIBs in your TIBTAB.
28	Device support for at least one of the 3270 family of terminals must be defined in your TIBTAB (for example, 3278).
32	Review your storage estimates for your general buffer pool.

**Comp. Operator** Action should be taken based on the return code received. Refer to the documentation for systems programmers for more information.

**OPC0073      \$1 \$2 invalid before LOGOFF/CANCEL**

**Explanation** The operator entered requested that a user be forced off the system. This command is invalid without a previous logoff attempt for the user in the case of a non-attached terminal or a previous cancel attempt for an attached task.

**System Action** The request is ignored.

**Comp. Operator** Before attempting to force a user, you must first attempt a 'LOGOFF' command for a non-attached user or a CANCEL command for an attached user. If this has been attempted and the user is still in the system, the force is accepted.

**OPC0074**      **\$1 \$2 - command invalid for attached user**

**Explanation**      The requested command was attempted for an attached user. This command cannot be issued for an attached user.

**System Action**      The request is ignored.

**Comp. Operator**      Attached users are a special case. However, another operator command that causes the effect you require is probably available. Determine what that command is and issue it for the user.

**OPC0075**      **No buffer available for OC wait list**

**Explanation**      The main SMARTS server environment task `OC` requires a storage area to build a list upon which it can wait. The storage for this list was not available.

**System Action**      &cmom terminates with an ABEND.

**Sys. Programmer**      Insufficient storage was available in the general buffer pool. Review the storage allocation for the SMARTS server environment generally and the general buffer pool in particular.

**OPC0076**      **Request denied. \$2 failed for \$1**

**OPC0077**      **DCB for COMPLIB load library chain has been closed**

**OPC0078**      **OPEN failed for COMPLIB**

**OPC0079**      **COMDMP - Dumpfile successfully opened**

**OPC0080**      **COMDMP - Dumpfile not opened**

**OPC0081**      **COMDMP - Dumpfile too small**

**OPC0082**      **COMDMP - Dump will be written to SYSLST**

**OPC0083**      **DUMP command operand must be either 'DISK' or 'NODISK'**

**OPC0084** \$1 \$2 \$3

**Explanation** The VSE trace facility (UPSI) has been activated.

**OPC0085** \$1 ready for communications

**OPC0086** TLINSP terminated

**OPC0087** Module '\$1' loaded for termination

**OPC0090** Invalid operand '\$1' for \$2 command

**Explanation** The operator command \$2 was issued with an operand of \$1. This operand is not valid for the command. When the string `\*no operand\*` appears, it indicates that the command was issued without an operand when an operand was required. For correct syntax, refer to information about the specified command.

**OPC0091** \$1 \$2 - \$3 \$4 cancelled successfully via ABEND

**Explanation** The command \$1 \$2 was issued by the operator. As a result, the program running on TID \$4 was terminated by ABENDING the program. This indicates that the program was associated and running on a task.

**System Action** The user program is abnormally terminated with a SMARTS server environment online dump.

**OPC0092** \$1 \$2 - \$3 \$4 cancelled successfully via POST

**Explanation** The command \$1 \$2 was issued by the computer operator. As a result, the user program running on TID \$4 was cancelled by posting the program active. This indicates that the program was associated with a task but was waiting on one or more ECBs to be posted active.

**System Action** The user program is abnormally terminated with a SMARTS server environment online dump.

**OPC0093** Task group \$1 does not exist

**Explanation** An attempt to display or modify the task group \$1 failed as the task group does not exist.

**Comp. Operator** Use the PG function of the UCTRL utility to determine what task groups are currently active on the system.

**OPC0094 Thread group \$1 does not exist**

**Explanation** An attempt to display or modify the thread group \$1 failed because the thread group does not exist.

**Comp. Operator** Use the TG function of the UCTRL utility to determine what thread groups are currently active on the system.

**OPC0098 Statistics print module (UCTRLP) could not be loaded**

**OPC0099 Command received at \$1 from \$2 was \$3**

**Explanation** This message is issued in response to any SMARTS server environment operator command. This command could have been issued from an operator console or generated internally during startup or termination.

**System Action** The command parsing and processing follows.

**Comp. Operator** This is an informational message only; no action is required.

**OPC0100 DYNALLOC (DEALLOCATE) failed, DDNAME=\$1 S99ERROR=\$2**

## **OUS — ";User" Operator Command Processor**

---

**OUS0001 Program \$1 attached, data '\$2'**

**Explanation** In response to the the `USER' operator command, the program \$1 was successfully attached with data \$2 as supplied in the operator command.

**OUS0002 Invalid program name**

**Explanation** The operator command USER specified an invalid program name to be attached.

**System Action** The USER operator command is terminated.

**Comp. Operator** For the correct syntax, refer to the information about the USER command.

**OUS0003 Invalid function**

- OUS0004**      **File name missing or invalid**  
**Explanation**      The operator command `USER ULIB...` was issued to request that certain files be closed; however, no files to be closed have been specified on the request, or the file names specified on the request are invalid.  
**System Action**      The request is terminated.  
**Comp. Operator**      Refer to the information about specifying file names on the `USER ULIB` request.
- OUS0005**      **Too many files (more than 5)**  
**Explanation**      The operator command `USER ULIB` was issued; however, the command specifies too many files at the one time. The maximum number of files that may be specified is currently 5.  
**System Action**      The request is terminated without closing any files.
- OUS0006**      **Program \$1 attached function \$2 DDN \$3**  
**Explanation**      The program \$1 was successfully attached to perform the \$2 function on file \$3.  
**System Action**      The attached program asynchronously attempts to complete the request.

## RES — Reentrant Program Support

---

- RES0002**      **Program \$1 not found**  
**Explanation**      This message is associated with the SMARTS server environment initialization of resident programs and indicates the program `\$1` is not in the SMARTS server environment ident STEPLIB, JOBLIB, or SYS1.LINKLIB libraries.  
**System Action**      The application program that was to be loaded as resident is not loaded.  
**Sys. Programmer**      Link the resident program, then restart the SMARTS server environment.  
**Appl. Programmer**      The program `\$1` is not loaded resident. If, at a later time, the program is linked to one of the above libraries and is subsequently loaded by an application program, it is loaded into the thread. If insufficient storage is available, the application program is abnormally terminated.
- RES0003**      **\$1 resident program load failed, code \$2**  
**Explanation**      This message is associated with the SMARTS server environment initialization of resident programs. The number indicated by CODE \$2 indicates the type of error and represents the return code from the operating system LOAD macro. Refer to the description of the LOAD macro's return code.  
**System Action**      The application program that was to be loaded as resident is not loaded.  
**Sys. Programmer**      Link the resident program, then restart the SMARTS server environment.

**Appl. Programmer** The program '\$1' is not loaded resident. If an application program subsequently loads program '\$1', it is loaded into the thread. If insufficient storage is available, the application program is abnormally terminated.

**RES0006 Program \$1 loaded - \$2 \$3**

**Explanation** Resident program \$1 was load successfully. For systems where modules can reside above the 16 Mb line, \$2 indicates if it was loaded 'ABOVE' or 'BELOW'. If the program is not reentrant, \$3 indicates that fact. Residentpage programs must be reentrant; otherwise, ABENDs occur when the program tries to alter itself.

**RES0007 \$1 programs loaded \$2, size \$3K**

**Explanation** This message indicates the number of resident programs that were loaded (\$1) and the total size of the resident programs (\$3). For systems where modules can reside above the 16 Mb line, \$2 indicates which 'set' of modules the message relates to; that is, one message is issued for the modules loaded below the line and one for the modules loaded above.

**Sys. Programmer** This information can be used to estimate the storage requirements of for the SMARTS server environment.

**RES0008 Duplicate resident program \$1 ignored**

**Explanation** The same program name \$1 was specified twice in the start-up parameters for the SMARTS server environment.

**System Action** Program \$1 is loaded once into the resident program area and the second request is ignored.

**Sys. Programmer** Remove one of the RESIDENTPAGE sysparms for the duplicated module.

**RES0011 Invalid command starting '\$1'**

**Explanation** The operand entered is not 'LOAD', 'DELETE', or 'REFRESH'.

**System Action** The command is ignored.

**Comp. Operator** Correct the command and reenter.

**RES0012 Invalid program name starting '\$1'**

**Explanation** The specified program name does not start with a letter or is more than eight characters long.

**System Action** The command is ignored.

**Comp. Operator** Correct the command and reenter.

**RES0013      Program \$1 already resident in SMARTS**

**Explanation**      A request was made to load a program that was already resident.

**System Action**    The command is ignored.

**Comp. Operator** Specify the `REFRESH` operand.

**RES0015      \$1 resident program \$2 successful \$3**

**Explanation** An operator request \$2 for resident program \$1 completed successfully. If the request involved the loading or refreshing of a program, \$3 indicates whether the program was loaded above or below the 16 Mb line and whether the newly loaded program is reentrant or not.

**RES0017      Program \$1 not found**

**Explanation**      A LOAD request failed because the specified program could not be found.

**System Action**    The command is ignored.

**Sys. Programmer** Place the specified program in the SMARTS server environment STEPLIB/LIBDEF.

**Comp. Operator** Notify the SMARTS server environment system programmer.

**RES0019      No storage available for loadpcb**

**Explanation**      A request was received to load a module into the resident program area. The SMARTS server environment attempted to acquire storage for a loaded program control block (LPCB) for the module and this request failed.

**System Action**    The request to load the program is terminated without the program being loaded.

**Sys. Programmer** In a case where storage is not available for the LPCB, it is unlikely that the storage will be available to load the actual module. In this case, the storage estimates must be reviewed to allow for situations where modules are loaded while the SMARTS server environment is running.

**RES0020      Program \$1 not resident**

**Explanation**      A DELETE or REFRESH request specified a program that was not resident.

**System Action**    The command is ignored.

**Sys. Programmer** Investigate and correct the error condition.

**Comp. Operator** Notify the SMARTS server environment system programmer.

**RES0021 Program \$1 not resident**

**Explanation** A request was made for action against resident program \$1; however, the SMARTS server environment has determined that this program does not exist in the resident program area.

**RES0022 Logic error in TTOCRP**

**Explanation** A call to the SMARTS server environment resident program search module resulted in an unexpected return code.

**System Action** The request is terminated.

**Sys. Programmer** Contact your technical support representative with the details of the exact command issued that caused the problem.

## ROL — Rollout / Rollin Processing

---

**ROL0002 TID \$1 no rollout slots available**

**Explanation** When a user program must be rolled out of thread, the SMARTS server environment first attempts to acquire space in the roll buffer to move the image in storage. If no space or roll buffer is available, the SMARTS server environment attempts to allocate space on the roll dataset(s) to roll the user out to disk. In this case, not enough contiguous space was available on any of the roll datasets.

**System Action** The SMARTS server environment continues processing. The user running the program for which the rollout failed is informed the next time they cause the SMARTS server environment to try to roll the program in again.

**Term. Operator** Too many terminals are in use at the same time. Wait a few minutes and try your request again. Contact the SMARTS server environment system programmer about expanding the SMARTS server environment rollout files so that more terminals can be in use at the same time.

**Sys. Programmer** Every program on every level requires space to which it can be rolled, either in the roll buffer or on the roll datasets. This space is only allocated when the program is rolled out and is freed when the program is rolled in again. Refer to the information about estimates for the roll subsystem.

**ROL0008 TID \$1 thread relocation failure**

**Explanation** During rollout processing, the thread for a relocatable program is prepared so that it can be rolled back into a different thread. If an error occurs during this relocation of the thread during rollout, this is remembered for the thread. When the thread is rolled back in again for whatever reason or in whatever thread, a check is made to see if the relocation on rollout worked. In this case, the relocation failed.

**System Action** The program is terminated with a dump and a message is sent to the user of the program.

**Term. Operator** Report this application or system error to the help desk or operations area.

**Sys. Programmer** An application overwrite in the thread corrupted control blocks specific to the SMARTS server environment that are necessary for relocation. One example of this would be the SMARTS server environment free queue element (FQE) chain. If you find no evidence of corruption, report the problem to your technical support representative, providing the thread dump to assist in the diagnosis of the problem.

**ROL0009      TID \$1 logic error during \$2**

**Explanation** A logic error during \$2 processing was encountered in the roll subsystem for tid \$1.

**System Action** The user is informed about the error, and where applicable, the program is terminated with a dump.

**Term. Operator** Report this system error to your help desk or operations area.

**Sys. Programmer** Something is wrong in the SMARTS server environment roll subsystem logic. \$2 indicates where this is; however, because of when the roll subsystem is called, the error may not be due to tid \$1, although that terminal is obviously directly affected by the problem. Report this to your technical support representative, providing the SMARTS server environment job log and any dumps produced for diagnostic purposes.

**ROL0010      TID \$1 no roll buffer space was available at time of**

**Explanation** This message is associated with ZRR00002 in that when the application program was last rolled out, the system was unable to find space on the roll datasets. An event has triggered an attempted rollin of the program; however, as the rollout failed, this message is sent to inform the user.

**System Action** The user program is terminated and the user is informed.

**Term. Operator** The installation defined insufficient resources for the SMARTS server environment. Report the problem to your help desk or operations area.

**Sys. Programmer** Review the space calculations for the roll subsystem.

**ROL0011      TID \$1 program cannot start, roll-out failed for**

**Explanation** If a user program is attempting to start, the previous thread user must first be rolled out of thread so that the new program can run. In this case, the rollout failed and rather than terminate a running session, the initialization request for the new program is rejected.

**System Action** The requested program is not started and the requestor is informed.

**Term. Operator** Problems exist with the SMARTS server environment roll subsystem that will be obvious to the systems programmer. Contact your help desk or operations area.

**Sys. Programmer** This error will be issued due to a previous rollout error. This can be determined from messages sent in conjunction with this message.

**ROL0012**      **TID \$1 thread image / TID / level mismatch**  
**Explanation**      The SMARTS server environment has rolled in a copy of a user application program. When the rolling completed, the tid and level of the program rolled in did not reflect what was expected.  
**System Action**      The program is terminated with a dump.  
**Term. Operator**      Report this system error to your help desk or operations center.  
**Sys. Programmer**      This is an internal logic error in the SMARTS server environment. Collect the SMARTS server environment job log and the dump and contact your technical support representative.

**ROL0013**      **TID \$1 thread mismatch at roll-in**  
**Explanation**      The SMARTS server environment rolled a user program back into a thread; however, the number of the thread is not what the TIB expected.  
**System Action**      The program is terminated with a dump and the user is informed.  
**Term. Operator**      Report this system error to your help desk or operations center to ensure they are aware of the problem.  
**Sys. Programmer**      The SMARTS server environment has an internal logic error. Collect the dump and SMARTS server environment job log and contact your technical support representative.

**ROL0014**      **TID \$1 roll-out failed, logic error**  
**Explanation**      This message is associated with ZRR00009. The logic error occurred while the tib was being rolled out. An event caused a rolling request and the opportunity is taken to tell the user about the problem rolling the program out.  
**System Action**      The user program is `terminated' and the user informed.  
**Term. Operator**      Report this system error to your help desk or operations area to ensure that they know the problem exists.  
**Sys. Programmer**      Refer to the previously issued ZRR00009 as correcting this problem will bypass this message being sent.

**ROL0020**      **TID \$1 active VSAM request detected at roll-out, file \$2**  
**Explanation**      When rolling out the contents of a thread, the SMARTS server environment detected an uncompleted request against the VSAM file indicated by \$2.  
**System Action**      Action depends on the type of request found active and on whether the SMARTS server environment requested that updates to the file \$2 be serialized.

1.	If the SMARTS server environment serializes updates for the file and the outstanding request is for update, a snap dump is taken, an ENDREQ issued for the request, and the application is terminated abnormally.
2.	In all other cases, processing continues. Treat the message as a warning about possible deadlock situations.

**Appl. Programmer** Generally speaking, applications designed for a multiuser environment like the SMARTS server environment should not issue terminal I/O or rollout operations while holding any VSAM resources.

**ROL0022 TID \$1 roll-out failed due to error in ROLL exit routine**

## RSM — Resource Management

**RSM0001 Resource pool \$1 created successfully**

**Explanation** A request to create a resource pool completed successfully. \$1 is the name of the resource pool created.

**RSM0002 Resource pool \$1 creation failed, rc=\$2 fdbk=\$3**

**Explanation** A request to create resource pool \$1 failed due to an error returned from the fixed length buffer pool manager. \$2 is the returned code from the fixed length buffer pool manager request and \$3 is the feedback code. See appendix H, Request Status , starting on page for detailed information.

**System Action** If the resource pool being created is the SMARTS server environment's general resource pool (that is, when \$1 is `GEN-RESR'), initialization of the SMARTS server environment fails. In other cases, the subsystem for which the resource pool is being created may not function correctly if it functions at all.

**Sys. Programmer** Based on the fixed buffer pool manager return and feedback codes, determine why the request failed and if possible correct the error. In most cases, the error is related to a shortage of storage. In cases where the problem does not appear to be related to the installation, report the error to the support area responsible for creating resource pool \$1.

**RSM0003 Resource pool \$1 deleted successfully**

**Explanation** A request to delete the \$1 resource pool completed successfully.

**System Action** The resource pool is no longer available for use.

**RSM0004      Resource Pool \$1 request \$2 rc=\$3 fdbk=\$4 reta=\$5**

**Explanation**      The resource manager detected a request that received a nonzero return code and a request that messages be issued. In this case the \$2 request against the \$1 resource pool received return code \$3 and feedback code \$4. See appendix H, Request Status , starting on page for detailed information. The address from where the request was issued is \$5.

**Sys. Programmer** Report this message indicating a logical error and the steps taken to create it to your technical support representative.

## STG — Storage Initialization

---

**STG0001      Insufficient storage at initialization for \$1**

**Explanation**      During initialization processing, storage was not available for the purpose indicated in the message. This value is self explanatory when it appears.

**System Action**      Depending on whether the storage is necessary or not, the SMARTS server environment may continue processing; however, it is possible that further problems with storage may occur later in the initialization process or the SMARTS server environment run.

**Sys. Programmer** Refer to the information about calculating the amount of storage required by the SMARTS server environment. Following this, adjust either the SMARTS server environment sysparm or the region size.

**STG0002      PAGE FIX failed, reply 'R' for Retry, 'E' for End**

**Explanation**      This message is associated with SMARTS server environment initialization processing in a virtual storage environment. The SMARTS server environment storage initialization routine was unable to page fix the threads or the tibtab due to a shortage of available real-page frames.

**System Action**      If R is replied to the outstanding message number, the page-fix will be retried. If E is replied to the outstanding message number, the SMARTS server environment initialization terminates with the message ZIM00003.

**Sys. Programmer** This is a severe error. Restart the SMARTS server environment when more real storage is available or decrease the size and/or number of threads.

**Comp. Operator** Reply R to retry the page-fix; E to terminate initialization of the SMARTS server environment.

**STG0004      Thread \$1 allocated \$2 below, \$3 above, Key \$4 \$5**

**Explanation**      This message is associated with main storage initialization of the SMARTS server environment. It is issued once for every thread being initialized. The value \$1 indicates the thread number. The values \$2 and \$3 indicate the storage size allocated to the thread below and above the 16 Mbyte line, respectively. The value \$4 indicates the storage protect key (in hexadecimal) assigned to the thread. \$5 indicates whether usage of the thread is limited.

**System Action**      Initialization of the SMARTS server environment continues.

**Sys. Programmer** The size, storage protect keys, and number of threads are determined by the THREADS, THSIZEABOVE, and PROTECTKEYS initialization parameters.

**Comp. Operator** This message is for information only; no action is necessary.

**STG0005 Storage size \$1K allocated for \$2 \$3**

**Explanation** This message indicates the amount of storage allocated for the purpose shown in the message. In a system where storage can reside above the line, the word (ABOVE) at the end of the message indicates that all the storage (or in some cases the majority of storage) was acquired above the 16 Mb line.

**STG0006 Error loading COMPAN/COMPAM; PAN interface disabled**

**STG0007 Subpool \$1 \$2 allocated E-sz \$3\$4 E-no \$5\$6 size \$7\$8 \$9**

**Explanation** A subpool with the number \$1 was allocated for the displayed buffer pool with the element size \$3\$4, element number \$5\$6, and total size \$7\$8. In a system where storage can reside above the line, if the buffer subpool is allocated above the line, 'ABOVE' appears at the end of the message.

**STG0008 WTO subsystem is active**

**Explanation** An active WTO table was found and will be used (XA only).

**System Action** Initialization continues.

**STG0009 Subsystem \$1 initialization complete**

**STG0010 Subsystem initialization failed, entry '\$1' not found**

**STG0011 WTO table initialization successful**

**Explanation** Under MVS/XA and above levels, a subsystem entry for the SMARTS server environment called 'COMP' is installed in the system. The subsystem entry was found and all associated processing completed successfully.

**STG0012 Unable to locate SMARTS subsystem entry '\$1'**

**Explanation** Under MVS/XA and above levels, a feature is available in the SMARTS server environment to provide more detailed information in the UQ M display. If this is required, a subsystem \$1 must be defined to the operating system. The named subsystem could not be found and therefore the facility is not active.

**System Action** Extended information is not available, so the standard copy of the master console is presented when the UQ M function is used.

**Sys. Programmer** Ignore this message unless you wish the extended feature within UQ M to be active. If you want it active, check that the subsystem name is not misspelled in the MVS definitions and that MVS actually built a subsystem entry.

## **SVR — Server Processing**

---

**SVR0001 Server \$1 Invalid command received**

**SVR0002 Server \$1 Control-Block address error**

**SVR0003 Server \$1 Getmain for Control-Block failed**

**SVR0004 Server \$1 Missing parameter**

**SVR0005 Server \$1 Missing numeric value**

**SVR0006 Server \$1 Class code invalid**

**SVR0007 Server \$1 Control-Block not initialized**

**SVR0008 Server \$1 Selection failed, parameter error**

**SVR0009 Server \$1 Command failed, parameter error**

**SVR0010 Server \$1 Invalid command for Server-Type**

**SVR0011 Server \$1 Unknown command received**

SVR0012 Server \$1 QPUT-Function failed

SVR0013 Server \$1 Server-PGM load failed

SVR0014 Server \$1 Register failed

SVR0015 Server \$1 Deregister failed

SVR0016 Server \$1 Delete PGM failed

SVR0017 Server \$1 Return code \$2 from INIT/TERM PGM

SVR0018 Server \$1 Function: \$2 SDE: \$3

SVR0019 Length: \$1 DAT: \$2

SVR0020 No Server directory

SVR0021 Server \$1 not defined

SVR0022 Server \$1 not initialized

SVR0023 No request entry

SVR0024 Server \$1 already active

SVR0025 Error loading TLINSERV

SVR0026 Server \$1 started

SVR0027 Server \$1 terminated

## TIB — Terminal Initialization

---

**TIB0005**      **Not enough storage available to build TIBTAB**

**Explanation**      This message is associated with SMARTS server environment TIBTAB initialization. Insufficient space is available in the region or partition for the TIBTAB.

**System Action**      Initialization of the SMARTS server environment is abnormally terminated.

**Sys. Programmer**      Increase the size of the region or partition.

**TIB0006**      **TIBTAB \$1 \$2**

**Explanation**      TIBTAB processing is complete. If the TIBTAB was loaded, the following appear:

TIBTAB tttttttt LOADED

— where `ttttttt' is the TIBTAB to be loaded.

If the TIBTAB is built dynamically, the following appear:

TIBTAB DYNnnnnn BUILT

— where `nnnnn' is the number of tibs to build.

**TIB0007**      **TIBTAB \$1 not found in library**

**Explanation**      This message is associated with SMARTS server environment TIBTAB initialization. The requested TIBTAB was not found in any library.

**System Action**      Initialization of the SMARTS server environment is abnormally terminated.

**Sys. Programmer**      Specify a valid TIBTAB name in the TIBTAB start-up parameter.

**TIB0009**      **Contents of TIBTAB \$1 invalid**

**Explanation**      This message is associated with SMARTS server environment TIBTAB initialization.

The probable cause is an attempt to run an earlier version of the SMARTS server environment TIBTAB instead of a tibtab for the current version of the SMARTS server environment.

Another possible cause is that the module is not a TIBTAB.

**System Action**      Initialization of the SMARTS server environment is abnormally terminated.

**Sys. Programmer** Check the last assembly and link of the specified TIBTAB.

**TIB0010      LOAD failed for TIBTAB '\$1'**

**Explanation**      This message is associated with the initialization of the SMARTS server environment. The requested load for the TIBTAB failed.

**System Action**    Initialization of the SMARTS server environment is terminated.

**Comp. Operator**   Execute the SMARTS server environment again. If this fails, notify the SMARTS server environment system programmer.

**TIB0014 Dynamic TIBTAB initialization completed**

**TIB0015 Dynamic TIBTAB initialization aborted: TIBTAB exhausted**

**TIB0016 There are no dynamic TIB updates**

**TIB0017 TIB \$1 allocated to \$2**

**TIB0018 TIB \$1 not available, dynamic definition \$2 skipped**

**TIB0019 TIB \$1 (\$2) data replaced by override definition**

**TIB0020 TIB definition \$2 skipped, already existing with TID \$1**

## **TMR — Timing Services**

---

**TMR0001      Program \$1 cancelled due to CPUTIME exceeded**

**Explanation**      Each time an application program is dispatched by the SMARTS server environment, it is given a certain amount of CPU time in which to complete its transaction and write a reply to the terminal. If an application program exceeds this amount of time, this message appears. This condition may be caused by an indefinite loop in application program '\$'.

**System Action**    The user program is abnormally terminated with a SMARTS server environment online dump.

A program exceeded the set thread time (probably because it was looping). If the program needs more time than is allowed, the time may be extended by using the ROLLOUT function within the program.

**Term. Operator** Contact the application programmer responsible for the program in use when the error occurred.

A program exceeded the set thread time (probably because it was looping). If the program needs more time than is allowed, the time may be extended by using the ROLLOUT function within the program.

**Sys. Programmer** The CPUTIME= initialization parameter of the SMARTS server environment is used to set the CPU time limits for each thread. If this parameter is not specified, the CPU time limit is set to two seconds for each thread.

A program exceeded the set thread time (probably because it was looping). If the program needs more time than is allowed, the time may be extended by using the ROLLOUT function within the program.

**Appl. Programmer** The CPU time limit is set for each thread when the SMARTS server environment is initialized.

A program exceeded the set thread time (probably because it was looping). If the program needs more time than is allowed, the time may be extended by using the ROLLOUT function within the program.

**TMR0002 Program \$1 cancelled after elapsed time exceeded**

**Explanation** Each time an application program is dispatched by the SMARTS server environment, it is given a certain amount of elapsed time in which to complete its transaction and write a reply to the terminal.

If an application program exceeds this amount of time, the computer operator is informed by either the TMR0003 or TMR0004 messages, but the application program is not automatically terminated by the SMARTS server environment because there is no way of ensuring that the application program is responsible for the elapsed time being exceeded. For example, the CPU could have been in STOP mode or a SYSTEM-MUST-COMPLETE function could be transpiring in another region.

The computer operator may choose to cancel the program once it has been determined that the problem is being caused by the application program and not by other circumstances.

This message appears when the computer operator has cancelled the application program by entering the CAN command.

**System Action** The program is abnormally terminated with a SMARTS server environment online dump.

**Term. Operator** Contact the application programmer responsible for the program in use when the error occurred.

**Appl. Programmer** The program was using more than its share of time in the thread. If the program needs more than 1 or 2 seconds of elapsed time in the thread, use the ROLLOUT function within the program to periodically relinquish the thread resource.

- TMR0003**      **Pgr \$1 time exceeded by \$2 sec. Tid=\$3 Thread=\$4/\$5**
- Explanation**      Each time an application program is dispatched by the SMARTS server environment, it is given a certain amount of elapsed time in which to complete its transaction and write a reply to the terminal. The amount of time is set by a SMARTS server environment initialization parameter. If an application program exceeds the set amount of time, this message appears. The program name is indicated by \$1; \$2 is the amount of excess time that has elapsed; the terminal identification number of the terminal with which the program is in conversation is indicated by \$3; \$4 is the thread group and \$5 is the subgroup name where the program is running.
- System Action**      No action is required. The SMARTS server environment cannot automatically cancel the application program because the application program may not be responsible for the elapsed times being exceeded (for instance, the CPU could have been in STOP mode or a SYSTEM-MUST-COMPLETE function could have been transpiring in another region).
- Sys. Programmer**      The REALTIME= initialization parameter of the SMARTS server environment is used to set the elapsed time limits for each thread. If this parameter is not specified, the elapsed time limit is set to seven seconds for each thread.
- Comp. Operator**      Take the necessary action to determine if the application program is responsible for the time being exceeded. If so, enter the operator CAN command to cancel it. Refer to the information about the CAN command.
- 
- TMR0004**      **Pgm \$1 time exceeded, Tid=\$2 Thread=\$3/\$4**
- Explanation**      Each time an application program is dispatched by the SMARTS server environment, it is given a certain amount of elapsed time in which to complete its transaction and write a reply to the terminal. The amount of time is set by a SMARTS server environment initialization parameter. If an application program exceeds the set amount of time, this message appears. The program name is indicated by \$1; the terminal identification number of the terminal with which the program is in conversation is indicated by \$2; \$3 is the thread group and \$4 is the subgroup name where the program is running.
- System Action**      No action is required. The SMARTS server environment cannot automatically cancel the application program because the application program may not be responsible for the elapsed times being exceeded (for instance, the CPU could have been in STOP mode or a SYSTEM-MUST-COMPLETE function could have been transpiring in another region).
- Sys. Programmer**      The REALTIME= initialization parameter is used to set the elapsed time limits for each thread. If this parameter is not specified, the elapsed time limit is set to seven seconds for each thread.
- Comp. Operator**      Take the necessary action to determine if the application program is responsible for the time being exceeded. If so, enter the operator CAN command to cancel it. Refer to the information about the CAN command.

**TMR0005 User=\$1 Tid=\$2 LU=\$3 autologoff time exceeded**

**Explanation** User \$1 running on tid number \$2 luname \$3 is logged off by the SMARTS server environment as **ENTER** has not been pressed at the terminal within the time specified in the **AUTOLOGOFF** sysparm of the SMARTS server environment.

**System Action** The system attempts to log the user off.

**TMR0010 Userid \$1: \$2**

**Explanation** A scheduled request was performed by the timer monitor for user ID \$1.

**TMR0011 Timer monitor active on Tid \$1**

**Explanation** The timer monitor is working correctly and attached TID \$1.

**TMR0012 Timer monitor stopped, error loading UTMEX2**

**Explanation** While attempting to load the exit UTMEX2, an error was encountered.

**System Action** The UTIMER monitor program UTIMRM is terminated.

**Sys. Programmer** Determine what caused the error and correct it.

**TMR0013 Timer monitor stopped, not enough storage to load**

**Explanation** Insufficient storage was available to load the user exit UTMEX2 into the thread for UTIMER.

**System Action** The UTIMER monitor program UTIMRM is terminated.

**Sys. Programmer** Increase the region size for UTIMRM with ULIB.

**TMR0014 Timer monitor stopped, already active on another Tid**

**Explanation** During startup, UTIMRM established that the monitor program was already active on another terminal in the system.

**System Action** UTIMRM terminates.

**TMR0020 Timer monitor message**

**TMR0021 Timer monitor message**

**TMR0022 Timer monitor message****TMR0031 User exit UTMEX3 gave invalid return code**

**Explanation** The UTIMRM exit UTMEX3 was invoked and returned an unexpected return code.

**System Action** The return code is ignored and execution continues.

**Sys. Programmer** UTMEX3 gave a return code which was logically not expected at that point by U2TSUB, a subroutine of UTIMRM. Correct the error in the exit causing the bad return code to be returned. Refer to the information about return codes expected from UTMEX3.

**TMR0032 `SYSJOBS' DD statement missing**

**Explanation** UTIMRM attempted to honor a request from UTIMER to submit a job; however, the SYSJOBS DD/DLBL statement could not be found and therefore the job could not be found to submit.

**System Action** The request to submit the job is ignored.

**Sys. Programmer** The jobs to be submitted by the timer monitor are expected in a dataset referenced by a SYSJOBS DD statement in the SMARTS server environment start-up procedure. Ensure that this DD statement exists.

**TMR0040 JIM initialization is not available**

**Explanation** The UTIMRM monitor program attempted to submit a job but discovered that the SMARTS server environment JES interface had not been successfully initialized and is therefore not available.

**System Action** The job is not submitted to the JES.

**Sys. Programmer** If this processing is required, ensure that a JES is available by using the JES sysparm and that it manages to initialize successfully.

**TMR0041 JIM operation for job \$1 failed, error code \$2**

**Explanation** UTIMRM attempted to submit job \$1; however, during the submission process, the &Cmon/JES interface module returned an unexpected return code \$2.

**System Action** Submission of the job is terminated.

**Sys. Programmer** A timer monitor JES operation could not be performed. Contact your technical support representative with details of the error.

**TMR0042**      **JIM request failed, function not supported**  
**Explanation**      UTIMRM attempted to use a function of the SMARTS server environment JES interface that is not supported.  
**System Action**      The request is aborted.  
**Sys. Programmer**      This message results from a return code 12 from the JES interface module. It could not perform the requested action.

**TMR0043**      **Requested spool function \$1 not supported**  
**Explanation**      UTIMRM requested that the SMARTS server environment JES interface module issue a \$1 request; however, the JIM did not support this.  
**Sys. Programmer**      This message results from a return code 16 from the JES interface module. The spool action is not supported. Contact your technical support representative for more information.

## ZDM — COMDMP Dump Dataset Processing(VSE Only)

---

**ZDM0001**      **COMDMP \$1 error RF = x\$2 FDBK = X\$3**  
**Explanation**      This message is associated with the SMARTS server environment dump dataset. On return from VSAM macro \$1, an error was detected. \$2 contains the return code, \$3 the feedback code.  
**System Action**      The program is terminated.  
**Sys. Programmer**      See VSE/VSAM return and error codes.  
**Comp. Operator**      Contact your SMARTS server environment system programmer.

**ZDM0002**      **COMDMP successfully initialized**  
**Explanation**      This message is associated with the SMARTS server environment dump dataset. The dataset was successfully initialized.  
**System Action**      This message is for information only.  
**Comp. Operator**      No action is required.

**ZDM0003**      **Dump will be written to SYSLST**  
**Explanation**      This message is associated with the SMARTS server environment dump dataset. During initialization of the SMARTS server environment, an error was detected at the open of the SMARTS server environment dump dataset COMDMP. Messages ZDM0001 or ZDM0006 are written together with this message. Look for more information there.  
**System Action**      The initialization continues. In case of a SMARTS server environment ABEND, the dump is written to SYSLST.  
**Sys. Programmer**      Contact your SMARTS server environment system programmer to correct the error.

**ZDM0004      Dump file successfully opened**

**Explanation**      This message is associated with the SMARTS server environment dump dataset. The dumpfile was successfully opened and will be used in case of a SMARTS server environment ABEND. This message is written together with ZDM0005.

**Comp. Operator** This message is for information only.

**ZDM0005      Dump will be written to COMDMP**

**Explanation**      This message is associated with the SMARTS server environment dump dataset. The dumpfile was successfully opened and will be used in case of a SMARTS server environment ABEND. This message is written together with ZDM0004.

**Term. Operator** This message is for information only.

**ZDM0006      COMDMP dumpfile too small**

**Explanation**      This message is associated with the SMARTS server environment dump dataset. The dump file COMDMP is too small to contain a SMARTS server environment ABEND dump.

**System Action**      In case of a SMARTS server environment ABEND, the dump is written to SYSLST.

**Sys. Programmer** Define the SMARTS server environment dataset with a larger extent and initialize the dataset.

**Comp. Operator** Contact your SMARTS server environment system programmer.

**ZDM0007      COMDMP now dumping \$1**

**Explanation**      This message is associated with the SMARTS server environment dump dataset. It indicates the storage area (\$1) that is currently being written to the dump dataset during a SMARTS server environment ABEND.

**Comp. Operator** This message is for information only.

**ZDM0008      COMDMP dump \$1 successfully written**

**Explanation**      This message is associated with the SMARTS server environment dump dataset. Dump number \$1 was successfully written to the SMARTS server environment dump dataset.

**Comp. Operator** This message is for information only.

**ZDM0009      Dump \$1 not printed - highest record number was never set**

**Explanation**      This message is associated with the SMARTS server environment dump dataset. Dump number \$1 will not be printed because it was not successfully written to the dump dataset.

**System Action**      The program is terminated.

**Comp. Operator** This message is for information only.

**ZDM0010      COMDMP dump dataset not initialized**

**Explanation**      This message is associated with the SMARTS server environment dump dataset. The dataset was not initialized correctly and thus cannot be used by the SMARTS server environment.

**System Action**    In case of a SMARTS server environment ABEND, the dump is written to SYSLST.

**Sys. Programmer** Initialize the SMARTS server environment dump dataset with the utility TUDUBTCH.

**Comp. Operator**    Contact your SMARTS server environment system programmer.

**ZDM0011      COMDMP not enough GETVIS for data**

**Explanation**      This message is associated with the SMARTS server environment dump dataset. There is not enough partition GETVIS to write records to the SMARTS server environment dump dataset.

**System Action**    In case of a SMARTS server environment ABEND, the dump is written to SYSLST. If this message is issued by a TUDUBTCH, the program is terminated.

**Sys. Programmer** Increase the partition GETVIS area.

**Comp. Operator**    Contact your SMARTS server environment system programmer.

**ZDM0012      COMDMP invalid RBA for GET/PUT : \$1**

**Explanation**      This message is associated with the SMARTS server environment dump dataset. During GET/PUT for COMDMP, an invalid RBA was encountered. The RBA is \$1.

**System Action**    In case of a SMARTS server environment ABEND, the dump dataset is closed and the dump written to SYSLST. If this message is issued by TUDUBTCH, the program ABENDs with a dump.

**Sys. Programmer** Contact SMARTS server environment technical support for problem analysis.

**Comp. Operator**    Contact your SMARTS server environment system programmer.

**ZDM0013      COMDMP RECSIZE/CISIZE not correct**

**Explanation**      This message is associated with the SMARTS server environment dump dataset. The file was not defined with the correct parameters.

**System Action**    The dump dataset is not initialized.

**Sys. Programmer** Define the cluster using the correct allocation parameters. (see example JCL in JCLINST8.J on the SMARTS server environment source library).

**Comp. Operator**    Contact your SMARTS server environment system programmer.

**ZDM0091 TUDUBTCH - Backup/Restore unsuccessful****ZDM0092 TUDUBTCH - Backup/Restore successful****ZDM0099 TUDUBTCH - invalid function**

**Explanation** This message is associated with the utility TUDUBTCH. The PARM value on the EXEC card contains an invalid function.

**System Action** The utility is terminated.

**Sys. Programmer** Refer to the information about allowed functions on the PARM statement.

**Comp. Operator** Contact your SMARTS server environment system programmer.

## ZLA — Security and Accounting

---

**ZLA0002 User \$2 is not authorized to invoke program \$1**

**Explanation** This message is associated with the SMARTS server environment program security routine. An installation may set security requirements on programs to prevent their unauthorized use. The appearance of this message indicates that the security requirements were not satisfied for the user \$1 for the program \$2.

**System Action** The request for the program is ignored.

**Term. Operator** The user ID may not be authorized to access the program that is being executed. Contact the individual responsible for the program you were using when the error occurred.

**Sys. Programmer** Refer to the information about SMARTS server environment program security.

## ZOS — OS Initialization

---

**ZOS0001 Authorization check failed**

**Explanation** The SMARTS server environment has checked to establish if it is running authorized and has discovered that it is not authorized.

**System Action** Initialization of the SMARTS server environment is aborted.

**Sys. Programmer** The SMARTS server environment must run authorized.

**ZOS0002      Dataset '\$1' not RECFM=U**

**Explanation**      The SMARTS server environment has opened the dataset \$1 which it expects to have a record format of undefined (RECFM=U). The dataset does not have this record format.

**System Action**    Initialization of the SMARTS server environment is aborted.

**Sys. Programmer** This will be issued in relation to the COMPINIT and COMPLIB dataset concatenations. As these must be load libraries, the datasets they point at must have RECFM=U.

**ZOS0003      LINK to program \$1 failed ABEND Code X'\$2'**

**Explanation**      During initialization, the SMARTS server environment issues MVS LINKs to other programs. In this case, the link to program \$1 terminated with ABEND code \$2.

**System Action**    Initialization processing of the SMARTS server environment continues, if possible.

**Sys. Programmer** Determine why the ABEND occurred using the ABEND code and correct the situation.

**ZOS0004      DDNAME '\$1' not found**

**Explanation**      During initialization processing, the SMARTS server environment attempts to open the COMPINIT and COMPLIB datasets. In this case, the DD \$1 was not found in the SMARTS server environment JCL.

**System Action**    If the dataset is COMPINIT, processing continues. If it is COMPLIB, initialization of the SMARTS server environment is aborted.

**Sys. Programmer** The COMPLIB DD statement at the very least must be specified in the SMARTS server environment JCL.

**ZOS0005      Unable to open '\$1' dataset**

**Explanation**      The SMARTS server environment attempted to open the \$1 datasets; however, the open failed.

**System Action**    If \$1 is COMPLIB, initialization of the SMARTS server environment is aborted. If it is COMPINIT, initialization processing continues.

**Sys. Programmer** The operating system normally issues messages related to this problem. These should help to determine and correct the problem.

**ZOS0006 Permanent I/O error on 'COMPLIB' dataset****ZOS0007      ESTAE failure RC '\$1'**

**Explanation**      During initialization, the SMARTS server environment attempts to establish an ESTAE recovery environment for the main task. The ESTAE request failed with return code \$1.

**System Action**    Initialization of the SMARTS server environment is aborted.

**Sys. Programmer** The return code from the ESTAE should describe why the request to set an ESTAE failed. Correct this error and retry.

**ZOS0009**      **\$1 failed for UCB \$2 code x'\$3'**

**Explanation**      The SMARTS server environment attempted to allocate or deallocate (as per \$1) the UCB identified by \$2; however, the operation failed. \$1 contains the error and information codes returned from OS DYNALLOC processing.

**Sys. Programmer**      Identify the cause of the failure as per the DYNALLOC error codes. This should generally only fail if the device is already allocated or the device is not eligible to be allocated. If the error falls out of one of the above categories, contact your technical support representative.

## ZTR — Trace Utilities

---

**ZTR0004**      **DYNALLOC Message Level changed to \$1 by user \$2**

**Explanation**      The value defined by SYSPARM DYNALLOC-MSGLEVEL changed.

**ZTR0005**      **\$1 Trace \$2 turned \$3 by user \$4**

**Explanation**      The trace class or option \$1 changed to the \$3 status by user \$4. \$2 indicates whether it is a trace class or option.

**ZTR0006**      **Trace TID changed to \$1 by user \$2**

**Explanation**      The TID number for which tracing is active changed to TID number \$1 by user \$2.

## ZTS — Thread Storage

---

**ZTS0001**      **Insufficient thread storage for request**

**Explanation**      The request for thread storage failed due to insufficient thread size.

**System Action**      The program is cancelled and an online dump is generated.

**Sys. Programmer**      Increase the region size for the program.

**ZTS0002**      **Attempt to free unallocated storage**

**Explanation**      An attempt was made to free thread storage not previously acquired.

**System Action**      The program is cancelled and an online dump is generated.

**Appl. Programmer**      Check the FREEMAIN requests for valid arguments.

**ZTS0003            Invalid Request/FQE detected**

**Explanation**        An error occurred in the free queue chain; for example, a free queue element (FQE) was overwritten.

**System Action**     The program is cancelled and an online dump is generated.

**Appl. Programmer** Check the free queue chain within the thread.

**ZTS0004            Bad request or FQE**

**Explanation**        An invalid thread storage request was made and an invalid free queue element (FQE) was detected.

**System Action**     The program is cancelled and an online dump is generated.

**Appl. Programmer** Validate the request and the FQE.

# A Allocated Error Numbers (errno)

---

Errno	C Macro Name	Description
1	E2BIG	Argument list too long.
2	EACCES	Permission denied.
3	EADDRINUSE	Address in use.
4	EADDRNOTAVAIL	Address not available.
5	EAFNOSUPPORT	Address family not supported.
6	EAGAIN	Resource unavailable.
7	EALREADY	Connection already in progress.
8	EBADF	Bad file descriptor.
9	EBADMSG	Connection already in progress.
10	EBUSY	Device or resource busy.
11	ECANCELED	Operation cancelled.
12	ECHILD	No child processes.
13	ECONNABORTED	Connection aborted.
14	ECONNREFUSED	Connection refused.
15	ECONNRESET	Connection reset.
16	EDEADLK	Resource deadlock would occur.
17	EDESTADDRREQ	Destination address required.
18	EDOM	Math argument out of domain function.
19	EDQUOT	Reserved.
20	EEXIST	File exists.
21	EFAULT	Bad address.
22	EFBIG	File too large.
23	EHOSTUNREACH	Host is unreachable.
24	EIDRM	Identifier removed.

Errno	C Macro Name	Description
25	EILSEQ	Illegal byte sequence.
26	EINPROGRESS	Connection in progress.
27	EINTR	Interrupted function.
28	EINVAL	Invalid argument.
29	EIO	I/O error.
30	EISCONN	Socket is connected.
31	EISDIR	Is a directory.
32	ELOOP	Too many levels of symbolic links.
33	EMFILE	Too many open files.
34	EMLINK	Too many links.
35	EMSGSIZE	Message too large.
36	EMULTIHOP	Reserved.
37	ENAMETOOLONG	File name too long.
38	ENETDOWN	Network is down.
39	ENETUNREACH	Network unreachable.
40	ENFILE	Too many files open in system.
41	ENOBUFS	No buffer space available.
42	ENODATA	No message available on the stream head read queue.
43	ENODEV	No such device.
44	ENOENT	No such file or directory.
45	ENOEXEC	Executable file format error.
46	ENOLCK	No locks available.
47	ENOLINK	Reserved.
48	ENOMEM	Not enough space.
49	ENOMSG	No message of the desired type.
50	ENOPROTOOPT	Protocol not available.
51	ENOSPC	No space left on device.
52	ENOSR	No stream devices.
53	ENOSTR	Not a stream.
54	ENOSYS	Function not supported.
55	ENOTCONN	The socket is not connected.
56	ENOTDIR	Not a directory.
57	ENOTEMPTY	Directory not empty.
58	ENOTSOCK	Not a socket.
59	ENOTSUP	Not supported. The implementation does not supported this feature of the standard.

Errno	C Macro Name	Description
60	ENOTTY	Inappropriate I/O control operation.
61	ENXIO	No such device or address.
62	EOPNOTSUPP	Operation not supported on socket.
63	E_OVERFLOW	Value too large to be stored in data type.
64	EPERM	Operation not permitted.
65	EPIPE	Broken pipe.
66	EPROTO	Protocol error.
67	EPROTONOSUPPORT	Protocol not supported.
68	EPROTOTYPE	Socket type not supported.
69	ERANGE	Result too large.
70	EROFS	Read-only file system.
71	ESPIPE	Invalid seek.
72	ESRCH	No such process.
73	ESTALE	Reserved.
74	ETIME	Stream ioctl() time-out.
75	ETIMEDOUT	Connection timed out.
76	ETXTBSY	Text file busy.
77	EWOULDBLOCK	Operation would block.
78	EXDEV	Cross device link.



# B SMARTS Server Environment Internal Macro Description

---

- ATTACH Function ..... 126
- CANCEL Function ..... 126
- CMTIMR Function ..... 127
- ABEXIT Function ..... 127

This document contains information on the SMARTS Server Environment Internal Macro.

## ATTACH Function

---

ATTACH is used to create new units of work in the SMARTS server environment

Return Code	Feedback Code	Description
0	>0	A new unit of work was created but did not initialize successfully. Check for other errors in the log to determine why the new unit of work could not initialize successfully.
4	-	Insufficient TIBs available in the system. The resources in the SMARTS server environment must be increased.
8	-	Program to be started was not found. This implies an error in the installation of SMARTS as the PAENSTRT module could not be loaded in the SMARTS instance.
12	-	Security error. This should not occur and should be reported to your Software AG technical support representative.
16	-	Invalid program name. This should not occur and should be reported to your Software AG technical support representative.
20	-	Insufficient space to create a new unit of work. The resources in the SMARTS server environment must be increased.

## CANCEL Function

---

CANCEL is used to cancel a unit of work in the SMARTS server environment

Return Code	Feedback Code	Description
>0	-	Insufficient space available to cancel the unit of work. The resources in the SMARTS server environment must be increased.

## CMTIMR Function

---

CMTIMR is used to add and delete internal timers in the SMARTS server environment

Return code	Feedback Code	Description
4	-	Requested storage queue not found. This should not occur and should be reported to your Software AG technical support representative.
8	-	Insufficient space available to set the timer. The resources in the SMARTS server environment must be increased.

## ABEXIT Function

---

ABEXIT is used to add and delete termination exits in the SMARTS server environment

Return code	Feedback Code	Description
4	-	No previous exit found. This should not occur and should be reported to your Software AG technical support representative.
8	-	Insufficient space available to set an exit. The resources in the SMARTS server environment must be increased.
12	-	Invalid parameters. This should not occur and should be reported to your Software AG technical support representative.



# C Request Status Codes

---

- Return Codes ..... 130
- Feedback Codes ..... 130

The 'status' of a request is returned in a fullword in its parameter list. The first halfword is a return code indicating the processing results of the request. The second halfword is a feedback area where the exact reason for a processing failure is returned.

This document covers the following topics:

## Return Codes

---

00	The request was processed successfully without problems. Any data returned by the request should be in the appropriate field in the parameter list.
04	The request was processed successfully; however, an unusual event occurred that is described by a feedback code. Any data returned by the request should be in the appropriate field in the parameter list.
08	The request was not processed. Additional information is provided in the feedback code.
12	The request was not processed; an error in the request itself or an error in the data provided for the request made it invalid. Additional information is provided in the feedback code.
16	The request may or may not have been processed: this cannot be determined due to a logic error encountered in the resource manager. Additional information is provided in the feedback code. This return code indicates a problem with the resource manager itself and needs to be reported to your local technical support representative.

## Feedback Codes

---

04	The request field of the parameter list contains an invalid or unknown request number.
08	The request did not include the address of a resource pool token area, which is required for all requests.
12	The request included the address of a resource pool token area; however, the indicated area was either nonzero for a 'Create' request or zero for any other request. When a resource pool is created, the nonzero token is provided. A pointer is maintained from the created header to the provided token to ensure that the same token area is provided on each request for a specific resource pool. This feedback code is returned when a zero token area is returned for a non-'Create' request indicating that the token area provided is not correct.
16	The request was issued without the 'Rname' parameter when it was required, as it is for all requests except the 'Freeall' request.
20	The resource pool is quiescing. An 'Erase' request was issued to delete the resource pool, which requires all resource manager users to complete their last request before the resource pool is deleted and the token returned to a free status.
24	A logic error was encountered while processing the request. This generally indicates that something later in the code was inconsistent with something earlier in the code.
28	The level of the macro used to build a parameter list for the resource manager is maintained in the parameter list. This feedback code indicates that the macro level set in the parameter level is incompatible

	with the level of the resource manager being used. Either the parameter list has been overwritten, or it was generated with a higher level macro that the resource manager supports.
32	A buffer pool error occurred. The resource manager uses the fixed buffer pool manager to acquire storage for the resource pool header. If the buffer pool token provided on the create request was invalid, the storage cannot be acquired. The resource manager also uses the fixed buffer pool manager to create a buffer pool for the resource pool being built. This may fail for various reasons, the most common of which is a shortage of storage in the region. A message with more details about the error is issued to the console with the return and feedback codes from the fixed buffer pool manager request.
36	When an `Add` request is issued, a buffer must be acquired to add the resource to the resource pool. The fixed buffer pool manager request to acquire a buffer failed because the maximum number of resources were already defined. Either the resource pool was filled with unknown resources or the maximum number of resources for the resource pool specified on the `Create` request was not sufficient.
40	When a `Get`, `Free`, or `Delete` request is issued, the necessary resource must be identified. A resource number was used to specify the resource but the number was either not allocated or invalid and therefore, the resource was not found.
44	To process an `Add`, `Delete`, `Get`, or `Free` request, the resource manager requires a resource record area. The request to acquire a resource record area from the subpool built during resource pool `Create` processing failed. As this subpool expands to satisfy requests, the failure indicates that sufficient storage is not available. Refer to the console log for any fixed buffer pool manager messages related to the buffer pool built by the resource pool `Create` request.
48	A user issued an `Add` request for a resource name/length combination that already exists in the resource pool.
52	The parameter indicating the number of resources to be defined in a resource pool was not specified for the `Create` request.
56	The maximum number of resources to be defined for a resource pool was specified on a `Create` request; however, this was lower than the base number of resources to be defined in the resource pool.
60	The resource manager requires a fixed buffer pool token to be provided on a `Create` request. This indicates that the token was not provided.
64	The `Chain` parameter was not specified when it was required, which it is for the `Add`, `Delete`, `Get`, and `Free` requests.
68	A resource was identified using a resource number with a name and length to verify that the resource number is correct. The resource number is a valid resource; however, the length provided to identify the resource and the current resource length are not equal.
72	This is returned in exactly the same way as for 68 except that the length matched but the provided name did not.
76	The requested resource was found; however, it is being deleted. When a `Delete` request is issued, the resource is only flagged as deleted until all users of the resource are finished with it. The resource is currently in this flagged state.
80	The resource manager encountered a logic error while trying to `Free` a resource record area.
84	A request was issued to `Free` a resource; however, the resource manager could find no record of the user having acquired the resource. This occurs when no previous `Get` request was issued for the resource, or a different `Chain` address was provided to the `Get` request than was provided to the `Free` request.

88	The resource manager has a record of the user holding the resource; however, the resource itself indicates that it is not held. This is a logic error.
92	A 'Get' was requested for a resource; however, the resource manager has determined that the caller already has the resource.
96	The 'Length' provided on a request to identify a resource is longer than the maximum resource length for the resource pool.
100	A logic error occurred while trying to free a resource buffer for 'Delete' processing.
104	A resource number was provided to identify a resource but the number is higher than the maximum number of resources in the resource pool.
108	A requested resource was not available. Because COND=YES was specified on the 'Get' request, the resource manager did not wait for the resource to become free.
112	A request was made to 'Free' a resource that has been deleted. This is a valid request; however, the status of the resource control block does not reflect the values expected under the circumstances. This is a logic error.