System7 Binary for Solaris - TCAP

Release Notes for Version 3.00

1. Overview

This release adds support for licensing of TCAP for a particular machine. In addition to this a fault has been cleared regarding the internal signal queue becoming exhausted when a large number of invokes are still active when the dialogue is aborted.

This release is fully backward compatible with the previous release. This is the first release since V2.00.

2. New functionality

2.1 Support for end user licensing

Prior to this release the use of Host Protocol Software required a signed licence agreement between the supplier and the end user.

This release offers the ability to license the software for a specific target machine using a licence file obtained electronically without the need to enter into a customer specific licence agreement.

To determine whether a licence file is required, run the binary using the –v command line option. If a licence file is required please refer to the *Host Protocol Licensing User Guide*. Note that a separate licence file is required for each target machine.

3. Faults cleared

3.1 Running out of internal signals

In previous software releases, if large numbers of invokes where active in a single dialogue at the same time when the dialogue was aborted or ended then an internal signal queue could become exhausted. This would result in a failure to free the excess number of the invoke structures. The previous releases would therefore only support up to 32 invokes per dialogue.

The exhaustion of the internal signal queue would be indicated by the module sending MGT_MSG_EVENT_IND (0x0008) message with a status value of ERR_NO_SDLSIG (0x2e hex, 46 dec). If the signal queue were low but not exhausted, then a MGT_MSG_EVENT_IND (0x0008) message with a status value of ERR_SDLSIG_LOW (0x2f hex, 47 dec) would be sent.

This new release prevents a separate internal signal being required for each invoke and therefore removes the limitations on the number of invokes per dialogue. There is no effect on the total number of invokes that can be simultaneously supported in the module.

Dialogic
25-Feb-03

Binary for Solaris - TCAP

Release Notes for Version 3.02

1. Overview

This is the first full release since V3.00. It introduces the ability to run the binary for a short period of time without needing to have a license so that it may be trialled. This mode is well suited to supporting development work done on hosts without valid licenses, prior to deployment on the target host(s).

This release also increases the maximum number of simultaneous dialogues that are supported. In addition, some minor problems have been resolved in this release.

Note: As a result of TCAP supporting a new parameter (see 2.4), it is essential that users of MAP, INAP and IS41 binaries insure that, at the same time upgrading TCAP, they upgrade MAP to version 3.02, INAP to version 2.02 and IS41 to version 2.01 (or above).

2. New functionality

2.1 Support for 64K dialogues

This release of the module increases the number of simultaneous dialogues and associated resources that can be supported by the module. The number of resources available in the old and new releases is shown below. These additional dialogues and invokes can only be used if the main TCAP configuration message (TCP_MSG_CONFIG) is changed to enable the larger number.

Resource	Old	New
Dialogues (Incoming and Outgoing)	16384	65535
Invokes	16384	65535
Temporary Dialogue Buffers (DBUFs)	256	256
Temporary Component Structures (CPT)	256	256

2.2 Trial mode

An additional command line parameter '-t' has been added. If this flag is set then the binary will run in trial mode and the presence of a valid license certificate will not be enforced. The binary will, however, terminate after ten hours. When used in trial mode, the module will create and execute one additional thread.

2.3 Module reset

A new message, TCAP Module Reset, has been defined. Sending this message to the TCAP module returns the module to an unconfigured state and allows a new TCAP Configure Module Request to be sent with new parameters. The advantage of this is that it allows TCAP configuration parameters to be changed without having to restart gctload.

The format of the message is shown below:

MESSAGE HEADER	
FIELD NAME	**MEANING**
type	**TCP_MSG_RESET (0x7786)**
id	0
src	Sending module_id
dst	**TCP_TASK_ID**
rsp_req	Used to request a confirmation
hclass	0
status	0
err_info	0
len	0

2.4 Protocol version support

This release supports the protocol version parameter as shown below:

Parameter	Mnemonic	Value (dec)
Protocol version	TCPPN_VERSION	20

For ANSI the format of the protocol version is defined in T1.114-1996 - a single octet where each bit relates to a particular protocol version.

For ITU the format of the protocol version follows the definition of a bitstring as defined in X.209. The low octet is used for the protocol version. The high octet is used to denote the number of bits not used in the low octet.

2.5 Transmission of User Information

Previous releases of TCAP would not support the inclusion of the User Information parameter within a U-ABORT indication for ANSI. TCAP has been changed so that User Information sent in U-ABORT indication messages can be recovered by the TCAP user.

2.6 Disable TCAP operation timer

The TCAP operation timer can be disabled (so that no timer is started for a specific operation) by specifying a len=0 for the TCPPN_timeout parameter.

Faults cleared

3.1 TCAP components in TC-UNI messages

In previous software releases transmitted unstructured TCAP dialogue messages (TC-UNI) could only contain INVOKE components. This release optionally allows the user to send any valid component in a TC-UNI message. Users wishing to send components other than Class 4 operation invokes should disable transmitted component filtering using the TCPF_DIS_TXFILT TCAP configuration option. This functionality permits TCAP Result components to be sent in TC-UNI messages. In order to receive these Result components the user must set the TCPF_DIS_RXFILT TCAP configuration option.

3.2 Recovery of messages encoded using indefinite lengths

Two problems with recovery of messages encoded using indefinite lengths have now been resolved. One affected messages containing error components, which would not be correctly recovered when the component was encoded using the indefinite length form. The other affected messages where the Source diagnostic dialogue portion field was encoded using an indefinite length. In each case, a message received from the network would be discarded and a TC-P-ABORT would be sent to the user.

3.3 Setting of tid_ninst parameter

The tid_ninst module configuration parameter is used to specify the number of bits that are used for the TCAP instance in the locally generated transaction ID. It is now possible for the number of bits to be zero where multiple instances of TCAP are not used. Previously, configuring the tid_ninst field to be zero resulted in incorrect operation where messages were sent to non-existent TCAP instances.

3.4 Timer for UNI operations

UNI operations do not expect any response or reject from the network. In previous release, a timer was started for UNI operations, and always expiring. This has now been fixed, and no timer is started for UNI operations (independently of TCPPN_timeout parameter). The dialogue is released once the invoke message has been sent to SCCP.

Binary for Solaris - TCAP

Release Notes for Version 3.03

1. Overview

This release introduces the Transaction State Replication feature and adds support for 5 bit sequence control. In addition, some minor problems have been resolved in this release.

This release is fully backwards compatible with the previous release.

2. New functionality

2.1 Transaction State Replication feature

The Transaction State Replication feature provides a method to make a master TCAP module send status information about the dialogues it currently owns to a backup TCAP module. The replicated status information allows the dialogues currently owned by the master TCAP module to be gracefully aborted by the backup TCAP module, either in case of failure or in case of planned maintenance of the system running the master TCAP module.

For further details, refer to the Application Note “Transaction State Replication” (GA234SSS).

2.2 Support 5 bits Sequence Control

In the previous software releases, when sequence control is used (bit 1 of the Indicator Octet of TCPPN_QOS parameter set) the sequence control key in the SCPPN_SEQ_CTRL parameter sent to SCCP was limited to 4 bits (i.e. had a range 0 to 0x0f). This could lead in some situations to inappropriate SLS values being generated and cause incorrect load balancing across links within a linkset.

In this release the sequence control key has been enhanced to 5 bits (i.e. a range from 0 to 0x1f), to provide the full range of the sequence control values supported by the SCCP module.

3. Faults cleared

3.1 TCP_MSG_CONFIG with a null parameter length

In previous software releases, the module would crash if receiving a TCP_MSG_CONFIG with a null parameter length. It now correctly handles this error condition.

3.2 TCP_MSG_RESET

Several minor issues related to TCP_MSG_RESET have been fixed in this release. In previous software releases, the TCAP module was not confirming a TCP_MSG_RESET received before TCP_MSG_CONFIG; TCP_MSG_RESET was also resetting the module id to 0. Finally, TCP_MSG_RESET was never released. TCP_MSG_RESET is now correctly handled at any time, independently of TCP_MSG_CONFIG; it is now correctly released, and does not reset the module id of the module anymore. The module id is unchanged by TCP_MSG_RESET.

3.3 Length Long Form

In the previous software releases, the module did not correctly handle the reception of an element whose length was badly coded in the long form. This error has now been corrected, and in this event the module will now send a Maintenance event TCPEV_SYNTAX_ERR.

3.4 Recovering corrupted transaction portions

In previous software releases, when a message containing a badly formatted transaction portion (e.g. a segment of an SCCP message) was received, this was handled incorrectly (and could occasionally cause the software to crash). Now, when a message containing a badly formatted transaction portion is received, the message is discarded and subsequently a TCAP Maintenance Event is reported.

3.5 Handling return error in TC-END

In previous software releasess, if TCAP received from SCCP a return error within a TC_END for a TCAP class 4 operation, this caused a component resource leak. This has now been fixed.

Dialogic
18-Jul-05

Binary for Solaris - TCAP

Release Notes for Version 3.04

1. Overview

This release extends the TCAP module to support for up to four Network Contexts. Each Network Context allows TCAP to support a different set of options when used in conjunction with multiple SCCP modules.

This release is fully backwards compatible with the previous release.

2. New Functionality

Network Context Configuration

A new configuration message TCAP_MSG_NC_CONFIG (0x7787) has been added to allow the configuration of additional Network Contexts. The TCP_MSG_CONFIG (0x7780) message should be used to configure the default Network Context for the first network. For each subsequent Network Context the TCP_MSG_NC_CONFIG message is required. The TCP_MSG_NC_CONFIG message contains parameters to define address format and TCAP specific options and therefore allows different behaviour for the module to be selected depending on the Network Context of the dialog. Section ‘Message handling’ below describes how this is selected using the TCPPN_NC parameter.

The meaning of the parameters in the TCP_MSG_NC_CONFIG message is the same as the equivalent parameters in the TCP_MSG_CONFIG message. When used to support multiple local point codes within the same network the options settings should typically be the same in both messages.

TCAP Message - TCP_MSG_NC_CONFIG

Synopsis:

Message used to configure the additional Network Context TCAP module for operation.

Message Format:

MESSAGE HEADER		
FIELD NAME		**MEANING**
type		**TCP_MSG_NC_CONFIG** (0x7787)
id		**Network Context id** (value 1 to 3)
src		Sending module_id
dst		TCP_TASK_ID
rsp_req		used to request a confirmation
hclass		0
status		0
err_info		0
len		40
PARAMETER AREA		
OFFSET	**SIZE**	**NAME**
0	1	**cnf_ver** - must be set to zero
1	1	**user_id** - TC-User module id
2	1	**nsap_id** - Network-layer module id
3	2	**flags**
5	2	**max_data**
7	1	**addr_format**
8	32	Reserved for future use, must be set to zero.

Network Context id

The Network Context id will identify the Network Context being defined. The default Network Context (0) is configured using the existing TCAP_MSG_CONFIG message therefore this message should only be used for Network Contexts 1 to 3. This assumes that four Network Contexts are permitted.

Note: All other Parameters have the equivalent meaning and types as those used in the TCP_MSG_CONFIG.

Two additional configuration status return values TCPE_INVALID_VERSION (9) and TCPE_INVALID_NC (8) have also been added to indicate inappropriate configuration using the TCP_MSG_NC_CONFIG message.

2.2 Network Context Message Handling

When a dialog is initiated remotely no change is required as TCAP will automatically determine which Network Context is appropriate. An indication of the Network Context for the dialog will be passed up to the user in the TCPPN_NC parameter. Where the dialog is initiated locally then the application needs to specify which Network Context the message is destined for. This also indicates which point code will be used as the originating point code.

The Network Context should be indicated in the first message for the dialog being used. In the case of TCAP this will be in the first TCAP Service Request, typically an Invoke Req, using the TCPPN_NC parameter.

If no Network Context is specified then the default Network Context NC0 is assumed.

Parameter name	TCPPN_ NC
Parameter length	Variable, typically 1. Length of zero indicates Network Context is unknown.
Parameter data	Network Context Identifier. Value of 0 should be used for the default NC defined by the TCP_MSG_CONFIG message. For other Network Contexts it should match that defined in the TCP_MSG_NC_CONFIG message.

Binary for Solaris - TCAP

Release Notes for Version 3.05

1. Overview

This release clears a fault in the TCAP Component resource handling that occurs in TCAP host binaries V3.03 to V3.04. Customers using these versions are advised to upgrade to this new release.

In other respects this release is fully backwards compatible with the previous release.

2. Faults Cleared

TCAP Component Resource Loss

Under some error cases such as the reception of a Return Error from a Class 4 operation a loss component buffer resource could be experienced. In this situation the user would then see the TCAP Software event TCPSWE_NO_TCPT. This release modifies the component handling to rectify this fault.

Dialogic
07-Feb-06

Binary for Solaris - TCAP

Release Notes for Version 4.00

1. Overview

This release expands TCAP to support Long Messages for the support of SCCP Segmentation and Reassembly procedures.

This is the first release since V3.05 and is backwards compatible with that release. This release is developed for use with the SS7 Development Package for Solaris V3.00 or later. It cannot be used with earlier development packages.

Customers who wish to make use of the Long Message support offered in V3.00 of the development package should upgrade to this release of software (Long Message support is required for SCCP Segmentation). In addition the customers should upgrade to SCCP for Solaris V3.00 or later. Other customers need not upgrade.

2 New functionality

2.1 Support for Long Messages

Starting with this release TCAP now supports the generation and reception of long SCPPN_USER_DATA in SCPPT_N_UNITDATA_REQ and SCPPT_N_UNITDATA_IND primitives to or from SCCP. Similarly, it supports the generation and reception of long TCPPN_COMPONENT parameters in the TCP_MSG_CPT_REQ and TCP_MSG_CPT_IND messages to and from the TCAP-User.

To enable correct operation the TCPF_SEGMENTATION (0x0100) flag of TCP_MSG_CONFIG (0x7780) must be set. The max_data field in the TCP_MSG_CONFIG (0x7780) message should also be increased and the GCT environment must be set up to handle long messages. If the max_data field value is not sufficient then the TCAP Maintenance Indication TCPEV_DATA_LEN_ERR will be generated.

See ga237sss.pdf for more details on the new encoding scheme allowing to support parameters longer than 255 bytes.

The TCAP module will fail to initialize if it is configured to support large messages (TCPF_SEGMENTATION flag set), but GCT has not been configured to support large messages.

2.2 Use of Solaris shared object

This release makes use of shared object version of the GCT library included in V3.00 of the development package.

Dialogic
31-May-06

Binary for Solaris - TCAP

Release Notes for Version 4.01

1. Overview

This release adds support for 8-bit Sequence Control Parameters and Destination Transaction IDs used in the routing of SUA messages are now supported.

This release also corrects the handling of the Application Context parameter when used with multiple Network Contexts. Users with configurations not utilising multiple Network Contexts need not upgrade.

This release is fully backwards compatible with the previous release.

2. New Functionality

2.1 Support for 8-bit Sequence Control Parameters

This TCAP release now supports the generation of 8-bit Sequence Control values in messages to SCCP (in the SCPPN_SEQ_CTRL parameter). This is to allow SCCP to make use of eight bits when generating SLS values.

2.2 Support for SUA

A new TCAP module option, TCPF_DTID_ON (0x0400), has been added to the Module Configuration Request message (0x7780). If the TCPF_DTID_ON option is set, this will enable messages to be routed using the Destination Transaction ID (DTID) which is required for SUA based configurations.

3. Changes

3.1 Application Context parameter handling for Network Contexts

The previous release would not correctly interpret the Application Context parameter (TCPPN_APPL_CONTEXT) when used with multiple Network Contexts. When handling this parameter the settings for Network Context 0 would always be used. This could lead to messages containing Application Context values which are valid in other Network Contexts to be rejected. This release correctly handles this parameter for all Network Contexts.

Binary for Solaris - TCAP

Release Notes for Version 5.00

1. Overview

This release is the first release since V4.01, it includes enhancements to the licensing mechanism to support new format tokens in license files. It introduces Dialogue Idle Timers to allow dialogues that have been left open for long periods to be aborted.

This release also corrects a number of problems relating to 8 bit SLS values, Dialogue Groups and the formatting of Abort messages sent to SCCP as detailed below. The release is fully backwards compatible with the previous release.

2. New Functionality

2.1 Dialogue Idle Timer added

An Idle Timer can now be set on dialogues. If the dialogue is inactive for a timeout period, either the dialogue is automatically aborted or the user is warned. The transmission or reception of dialogue messages will reset the timer.

A new optional parameter is added to allow Dialogue Idle Timers, TCPPN_DLG_IDLE_TIMEOUT. This parameter can be set by the user on TC-BEGIN or TC-CONTINUE messages.

Parameter name	TCPPN_DLG_IDLE_TIMEOUT (0x16)
Parameter length	Fixed, set to 2.
Parameter data	Length of idle timeout for Dialogue. Double octet, set to 0 or a number to specify the timeout duration in seconds. 0 = No timer (will cancel an existing timer)

The user can change the timeout for outgoing or incoming established dialogues by re-defining the parameter in the TC_CONTINUE message. The user can similarly cancel an established idle timeout by setting the parameter to zero or set a new idle timeout where no timeout was initially set.

The TCPPN_DLG_IDLE_TIMEOUT parameter can be specified as a Module Default parameter using the TCP_MSG_DEF_PARAM message. This allows an idle timeout to be set on all incoming dialogues from SCCP.

An option flag is added to the TCP_MSG_CONFIG message to control the use of Dialogue Idle Timeouts.

Value	Mnemonic	Description
0x0200	TCPF_DLG_TIM_ABORT	Enable automatic abort of dialogue after timeout.

The option flag controls what happens when a timer expires. If flag is reset (zero), the module just sends an event to the management module on the expired timer, the timer is then reset with the previous duration. The second time the idle timer expires the event is again reported but the timer is not reset. If the flag is set, the dialogue is automatically aborted when the timer expires.

When the above option is set to abort, a maximum of 10 dialogue aborts are allowed for each timer tick. When more than 10 dialogues timeout on the same tick, further dialogues over 10 will have their timeouts extended and spread out over the next 2 seconds. This prevents systems being overwhelmed by dialogues being aborted on the same tick. When the module starts to defer Idle Timeouts, it sends the TCPEV_EXCESSIVE_DLG_ABORTS event to the Maintenance module (see below).

Two new TCP_MSG_MAIN_IND events are added for Dialogue Idle Timeouts:

- When the Idle Timer expires the TCPEV_DLG_TIM_TIMEOUT event is sent to the management module notifying the user. This event is reported whether or not the dialogue is aborted.

- The TCPEV_EXCESSIVE_DLG_ABORTS event is sent when aborts resulting from Dialogue Idle Timers start to be deferred. This event will only be sent once per tick (the TCPEV_DLG_TIM_TIMEOUT event can be sent multiple times per tick).

Mnemonic	Code	ID	Description
TCPEV_DLG_TIM_TIMEOUT	8	dialogue_ID	The dialogues idle timer has expired.
TCPEV_EXCESSIVE_DLG_ABORTS	9	0	Idle Aborts are being deferred for this tick.

3. Changes

3.1 SCPPN_SEQ_CTRL values now use 8 bits

The previous release failed to send 8 bit SLS values to the SCCP module. This release sends 8 bit SLS values to SCCP using the

SCPPN_SEQ_CTRL parameter (if Sequence Control is required). The SLS value used is defined either directly by the TCPPN_QOS parameter from the TC-User (see TCAP Programmer's Manual) or internally by the TCAP module (from the least significant 8 bits of the Dialogue Reference).

Note: From this release, if Sequence Control is enabled, TCAP will always send SCPPN_SEQ_CTRL parameter values in the range 0x00 to 0xff to the SCCP module.

3.2 TC-UABORT uses Dialog Portion ABRT APDU

Previous releases failed to include an ABORT APDU Dialog Portion in the message sent to SCCP when the TC-User sends a TC-UABORT and TCAP module is in the 'Initiation Sent' state. This caused the receiving TCAP to reject the U-ABORT (the user was sent a P-ABORT instead). This release corrects the fault and sends an ABORT APDU Dialog Portion in the SCCP message when in this state.

3.3 Allow zero sized incoming or outgoing Dialogue Group ranges

The previous releases could fail if a Dialogue Group range was set to zero. This release allows either the incoming or the outgoing range to be set to zero. As before, if both ranges are set to zero the Dialogue Group is rejected.

3.4 Active dialogue count in Module Status Message for Dialogue Groups

For previous releases the Module Status message (TCP_MSG_R_MOD_STATUS, 0x6796) did not count of any active incoming dialogues allocated to Dialogue Groups in its returned 'num_ic_dlg' field. This release fixes this issue and the returned 'num_ic_dlg' field is now the total count of active incoming dialogues across the TCAP module and includes dialogues allocated to Dialogue Groups.

Dialogic
25-Sep-08
Updated 28-Nov-08

Binary for Solaris - TCAP

Release Notes for Version 5.01

1. Overview

This release adds enhancement to the configuration of dialogue ranges now allowing the full available dialogue range to be allocated in any way between incoming and outgoing dialogues ranges.

This release is fully backwards compatible with the previous release.

2. New Functionality

2.1 Enhancement of Dialogue Ranges

The configuration of dialogue ranges has been enhanced by removing the restriction that had previously required the most significant bit to be set for incoming user dialogues. TCAP no longer inspects whether the most significant bit is set in the 'base_icdlg_id' parameter (offset 18) in the TCAP Configuration Request message (TCP_MSG_CONFIG, 0x7780) and that all values for the configured outgoing range are less than 0x8000. This allows the full available dialogue range (0x0000 to 0xffff) to be allocated in any way between incoming and outgoing dialogues ranges.

The configured incoming and outgoing dialogue ranges can now start anywhere and be any size, except for the following restrictions:

- The incoming and outgoing ranges cannot overlap each other.
- Neither configured range should exceed the maximum possible dialogue number (0xffff).
- The incoming and outgoing range bases (base_ogdlg_id & base_icdlg_id) cannot be equal.
- The number of dialogues for <u>both</u> incoming and outgoing ranges cannot be set to zero, however it is possible to configure one of the ranges to have zero dialogues.
- The total number of dialogues for both incoming and outgoing ranges cannot exceed the maximum number of dialogues allowed by the module (normally 0xffff).

If any of the above conditions are not met, the TCP_MSG_CONFIG message will be rejected with status TCPE_BAD_PARAM (0x06).

Binary for Solaris - TCAP

Release Notes for Version 5.02

1. Overview

This release adds the ability to send trace messages to a nominated Trace module and corrects a problem with slow license validation on some systems.

This release is backwards compatible with the previous release.

2. New functionality

2.1 Add field for trace module id to configuration message

A Trace Module ID can now be defined in the TCAP Module Configuration message (TCP_MSG_CONFIG). The new '**trace_id**' field uses a previously reserved field in the message. If trace_id is defined as non-zero, the all trace messages are sent to that module id. If trace_id is set to zero, trace messages are sent to the Management Module ID (mngt_id) as for previous releases.

The format of the TCAP Module Configuration Message is now defined as –

Message Header		
Field Name	**Meaning**	
type	**TCP_MSG_CONFIG** (0x7780)	
id	0	
src	Sending module ID	
dst	**TCP_TASK_ID**	
rsp_req	used to request a confirmation	
hclass	0	
status	0	
err_info	0	
len	40	
Parameter Area		
Offset	**Size**	**Name**
0	1	**cnf_ver** - must be set to zero
1	1	Reserved, must be set to zero.
2	1	**user_id** - TC-User module ID
3	1	**nsap_id** - Network-layer module ID

4	1	**mngt_id** – Management module ID
5	1	**maint_id** – Maintenance module ID
6	2	**flags**
8	2	**nog_dialogues**
10	2	**nic_dialogues**
12	2	**num_invokes**
14	2	**num_components**
16	2	**base_ogdlg_id**
18	2	**base_icdlg_id**
20	1	**tid_ninst**
21	1	**tid_ndref**
22	1	**tid_nseq**
23	1	**tcap_instance**
24	2	**max_data**
26	2	**dlg_hunt**
28	1	**addr_format**
29	1	**trace_id** – Trace module ID
30	10	Reserved for future use - set to zero

trace_id

Module identifier defining the destination for all trace messages.

3. Changes

3.1 Slow license validation

Some users have experienced problems with slow license validation on certain platforms. Previous releases worked around this problem but required licenses to be reactivated. With this release the problem is corrected without the need to reactivate the license.

Dialogic
09-Apr-09