Binary for Windows - TCAP

Release Notes for Version 5.03

1. Overview

This release introduces the Transaction State Replication feature.

It resolves problems with the handling of badly coded long form elements and introduces support for 5 bit sequence control. In addition, some other 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).

3. Faults cleared

3.1 TCP_MSG_CONFIG with a null parameter length

In previous software release, the module would throw an exception 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 release, 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 previous software releases, the module did not correctly handle the reception of an element whose length is 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 portion

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 Support 5 bits Sequence Control

In 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 therefore 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.

3.6 Handling return error in TC-END

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

Dialogic
21-Apr-05

Binary for Windows - TCAP

Release Notes for Version 5.04

1. Overview

This release adds support for multiple Network Contexts. Each Network Context allows TCAP to support a different set of options when used in conjunction with multiple SCCP modules. When used with the SIU mode of the SS7G2x Signaling Server and the DTC module this functionality can be used to support multiple local point codes and mixed ANSI and ITU-T configurations.

In addition, this release clears a fault in the TCAP Component resource handling that was introduced in V5.03. Users of this version are advised to upgrade to this new release.

The release is fully backwards compatible with the previous release.

2. New functionality

2.1 Network Context Configuration

A new configuration message, TCP_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. 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 2.2 covers 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 identifies 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 (21)
Parameter length	Variable, typically 1. Length of zero indicates Network Context is unknown.
Parameter data	Network Context Identifier If the default NC is being used then this parameter is optional. If present it should have a value of 0. For other Network Contexts it should match the value defined in the relevant TCP_MSG_NC_CONFIG message.

3. Faults Cleared

3.1 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 causing the TCAP Software event with status TCPSWE_NO_TCPT to be reported. This fault, first introduced in V5.03 has been corrected.

Binary for Windows – TCAP

Release Notes for V6.00

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 V5.04 and is backwards compatible with that release. This release is developed for use with the SS7 Development Package for Windows V5.00 or later. It cannot be used with earlier development packages.

Customers who wish to make use of the Long Message support offered in V5.00 of the development package should upgrade to this release of software (Long Message support is required for SCCP Segmentation). 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 Windows DLL

This release makes use the DLL version of the GCT library included in V5.00 of the development package.

Dialogic
21-Jul-06

Binary for Windows - TCAP

Release Notes for Version 6.01

1. Overview

This release enhances the handling of TCAP timers to maintain TCAP dialogs and their status when the connection to a host is no longer available. In addition, 8-bit Signalling Link Selection (SLS) values 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 Enhanced timer handling

The timer functionality has been enhanced and now enables pending TCAP dialogs and their status to be maintained when the connection to a host is lost.

For each new dialog, a per-dialog timer will now be initiated only if it has been specified by the user. The timeout value can be set by including the new optional parameter, TCPPN_DLG_IDLE_TIMEOUT (0x16), in the TC-BEGIN message.

Parameter name	TCPPN_DLG_IDLE_TIMEOUT **0x16**
Parameter length	Fixed, set to 2.
Parameter data	Length of timeout for Dialogue. Double octet, set to 0 or a number to specify the duration for that Dialogue in timer ticks. 0 = No timer is run

A new TCAP module option, TCPF_DLG_TIM_ABORT (0x0200), has been added to the Module Configuration Request message (0x7780) to optionally determine the behaviour of the module when a timer expires. If this option is set, this will automatically close and subsequently abort the dialog on timer expiry.

If the TCPF_DLG_TIM_ABORT option is not enabled, the module will send a Maintenance Event Indication (0x07a1) with status 0x08, to

notify the user that the affected timer has expired. Additionally, the timer will be reset to the initial timeout value.

Support of 8-bit SLS values

This release extends support for 8-bit SLS values.

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.

Dialogic
15-Nov-07

Binary for Windows - TCAP

Release Notes for Version 6.02

1. Overview

This release adds enhancements 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.

In addition, this release also corrects the handling of 8-bit SLS values, Dialogue idle timers, 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 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).

3 Changes

3.1 Dialogue Idle Abort operation

In previous releases, a maximum of only 2 dialogues per 100ms tick would be aborted when checking for Idle Timeout. Further dialogues (which should have been aborted on the tick) were not aborted.

This release will abort a maximum of 10 dialogues per tick for Idle Timeout. 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 change corrects the fault and prevents systems being overwhelmed by dialogues being aborted on the same tick.

When the module starts to defer Idle Timeouts (as above), it will send a new event TCPEV_EXCESSIVE_DLG_ABORTS (0x0009) to the Maintenance module in a TCP_MSG_MAINT_IND message. This event will only be sent once per tick. (It should be noted that the maintenance event TCPEV_DLG_TIM_TIMEOUT (0x08) continues to be sent each time a dialogue is aborted.)

3.2 Changes to Dialogue Idle Timeout operation

The TCPPN_IDLE_DLG_TIMEOUT parameter has been changed to specify the Dialogue Idle Timeout value in <u>seconds</u> (the previous release used 100ms ‘ticks’). This brings the parameter in line with the other TCAP timeout parameter TCPPN_TIMEOUT.

The TC-User can now define the Dialogue Idle Timeout parameter TCPPN_DLG_IDLE_TIMEOUT when sending TCPPT_TC_CONTINUE for an established dialogue. This allows the user to change the timeout duration for outgoing or incoming dialogues. The user can also cancel an established idle timeout by setting the parameter to zero or set a new idle timeout where no timeout was initially set.

3.3 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 (refer to the 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.

3.4 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.

3.5 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.

3.6 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 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
03-Dec-08

Binary for Windows - TCAP

Release Notes for Version 6.03

1. Overview

This release adds the ability to send trace messages to a nominated Trace module.

The use of trial mode is now limited to one hour but in all other respects this release is backwards compatible with the previous release.

2. New functionality

2.1 Trace module id added 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. Other Changes

3.1 Trial Mode

When operating in trial mode (using the –t command line option), the period of operation is now restricted to one hour after which the binary will terminate.

Dialogic
28-Jan-10