**Binary for Windows - SCCP**

**Release Notes for Version 5.00**

# 1 Overview

This release expands SCCP to support Segmentation and Reassembly procedures.

This is the first release since V4.04 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.

This release also adds a number of configuration options to enhance compatibility with different SCCP networks. New optional functionality includes protocol class validation and forced use of XUDT messages.

The release also corrects a fault in the generation of UDTS/XUDTS messages in Network Congestion situations.

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 Use of Windows DLL

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

## 2.2 Segmentation Support

This release supports segmentation and reassembly. If SCPXF_SEGMENTATION bit (0x80) of the ext_option field of SCP_MSG_CONFIG is set, and if the GCT environment has been setup to handle large messages, the SCCP module will re-assemble segmented messages received from MTP3 prior to passing the resulting message up the stack to TCAP. Similarly, it will handle long messages from TCAP and segment them to MTP3.

Refer to Application Note ga237sss.pdf *Using SCCP Segmentation and Reassembly* for more details on the new encoding scheme allowing to support parameters longer than 255 bytes.

The existing option SCPXF_ONBEHALF_REASSEMBLY allowed user applications to perform reassembly but does not permit segmentation

itself. The segmentation support enabled via the new SCPXF_SEGMENTATION option offers a more complete solution for segmentation and reassembly. The two options cannot be cannot be used at the same time.

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

## 2.3 Additional error report in SCP_MSG_CONFIG confirmation

In order to simplify the identification of configuration errors SCP_MSG_CONFIG (0x7740) now supports one extra field, error_offset (SCPMO_CONFIG_error_offset=58).

The module ignores this field on reception of the SCP_MSG_CONFIG message but sets the field in the SCP_MSG_CONFIG confirmation (0x3740) if an error in the configuration is found. In this situation then the status field will also be set to a non-zero value.

The error offset field gives information about the byte offset of the parameter area in SCP_MSG_CONFIG which causes the configuration to fail. In order to do this the SCP_MSG_CONFIG message length must be long enough, i.e. at least 60 bytes. There is no change of operation when SCP_MSG_CONFIG length is less than 60 bytes.

## 2.4 Additional Information in SCCP Maintenance Events

In this software release, additional information may be added in the parameter area of the SCCP Maintenance Event Indication message (SCP_MSG_MAINT_IND 0x0761), giving more detailed information for the event including a cause value and additional parameters where appropriate.

The following is an example message parameter area showing the SCPEV_RTF_SS_CONG maintenance indication.

| **Parameter Area** | | |
|---|---|---|
| **OFFSET** | **SIZE** | **NAME** |
| 0 | 1 | Cause value |
| 1 | 3 | Point code of replicate sub-system (note least significant byte first as per Q.713) |
| 4 | 1 | Sub-system number |
| 5 | 1 | Message Priority |
| 6 | 1 | Congestion Level |

The table below lists all of the maintenance indications together with the list of parameters supported by that indication.

| Maintenance event code | | Len | Cause value | Additional parameters | | | |
|---|---|---|---|---|---|---|---|
| 1 | SCPEV_RTF_TTYPE | 1 | 0 | | | | |
| 2 | SCPEV_RTF_TVAL | 1 | 1 | | | | |
| 3 | SCPEV_RTF_NET_FAIL | 5 | 5 or 245 | PC | SSN | | |
| 4 | SCPEV_RTF_NET_CONG | 7 | 6 | PC | 0 | P | CL |
| 5 | SCPEV_RTF_SS_FAIL | 5 | 3 or 243 | PC | SSN | | |
| 6 | SCPEV_RTF_SS_CONG | 7 | 2 or 242 | PC | SSN | P | CL |
| 7 | SCPEV_RTF_UU | 5 | 4 | PC | SSN | | |
| 8 | SCPEV_RTF_SIF_ERR | 1 | 252 | | | | |
| 9 | SCPEV_RTF_TOO_LONG | 1 | 253 | | | | |
| 10 | SCPEV_RTF_UNKNOWN | 1 | 7 or 8 | | | | |
| 11 | SCPEV_STX_ERR | 0 | | | | | |
| 12 | SCPEV_SOR_GRANT | 0 | | | | | |
| 13 | SCPEV_SOR_DENY | 0 | | | | | |
| 14 | SCPEV_CONN_FAIL | 0 | | | | | |
| 15 | SCPEV_RESET_FAIL | 0 | | | | | |
| 16 | SCPEV_RESTART_IND | 0 | | | | | |
| 17 | SCPEV_RTF_CANNOT_ REASSEMBLY | 6 | 10 | PC | SSN | RS | |

Parameters:

| Data | Len | Description |
|---|---|---|
| Cause Value | 1 | See Cause Value table (below) |
| PC | 3 | Originating Point Code (LSB first) |
| SSN | 1 | Sub-system Number |
| P | 1 | Message Priority |
| RS | 1 | Remaining Segment counts |
| CL | 1 | Congestion Level |

Cause Values:

| Description | Value |
|---|---|
| No translation for an address of such nature | 0 |
| No translation for this specific address | 1 |
| Remote Subsystem congestion | 2 |
| Remote Subsystem prohibited | 3 |
| Unknown Local Subsystem | 4 |
| Unknown DPC | 5 |
| DPC congested | 6 |

| Description | Value |
|---|---|
| Translation Error | 7 |
| Hop Counter Failure | 8 |
| Destination cannot perform reassembly | 10 |
| Local Subsystem Congestion | 242 |
| Local Subsystem Prohibited | 243 |
| DPC Prohibited | 245 |
| SIF Error (Message too long for MTP3) | 252 |
| Message too long | 253 |

## 2.5 Support for SST in XUDT messages

Previous releases of the SCCP module only handled SST contained in UDT messages and not when contained in XUDT messages. The SCCP module now correctly handles SST contained in XUDT messages, sending SSA (in UDT) when the subsystem is available.

## 2.6 Called Party address point code handling for GTT relay

The default operation of the module is to add the point code from the MTP routing label when the calling party address does not contain a point code. The existing extended option SCPXF_NO_CALLING_PC allows this point code insertion to be prevented for received messages which terminate locally. This functionality has been extended and point code insertion for messages which are relayed after Global Title Translation can now also be prevented using the same extended option.

## 2.7 National bit handling for GTT

The handling of the national/spare bit in the SCCP called party address has been extended to offer two modes of operation. The default handling will take the value national/spare bit from the configured primary or secondary address. If the extended option SCPXF_IGNORE_GTT_NAT_IND (0x0800) is set then the value of the national/spare bit will be copied from the original pre-translated address. The handling of addresses which do not require translation is unchanged.

## 2.8 Protocol Class validation

The module can now optionally validate message types against SCCP protocol classes. Set the extended option SCPXF_VALIDATE_PCLASS (0x0400) in the SCP_MSG_CONFIG (0x7740) message to enable this functionality. Invalid messages will be rejected with a message SCP_MSG_MAINT_IND (0x761) and a status value of SCPEV_STX_ERR (0xb).

## 2.9 SCCP management support for SST

The mode can now optionally support the handling of received SST message for SCCP Management itself (sub-system 1). If the extended option SCPXF_SCMG_SST_SUPPORT (0x1000) in the SCP_MSG_CONFIG (0x7740) message is set then the module will respond with a SSA in response to a SST for the local point code and SSN 1.

## 2.10 RSS availability on MTP-RESUME

The default behavior for the module is to initiate a sub-system test (SST) procedure 30 seconds after an MTP-Resume has been received for a remote signaling point. The module now allows the user to turn off the SST procedure for a specific remote sub-system by setting an option in the **ssr_flags** field of the SCP_MSG_CNF_SSR (0x7741) message. This new option SSRF_RSS_AVAIL (0x0004) is only appropriate for remote sub-systems and has no affect on remote signaling points or local sub-systems.

## 2.11 Forced use of XUDT

The module has been enhanced to allow the use of XUDT messages to be forced used instead of UDT. If XUDT messages are required then the option SCPXF_FORCE_XUDT (0x0200) should be set in the extended options field of the SCP_MSG_CONFIG (0x7740) message.

The XUDT message will include the hop counter value specified in the hop_counter parameter in the SCP_MSG_CONFIG (0x7740) message. If this parameter is not specified then a default value of 15 (0xf) is used instead.

## 2.12 UDTS/XUDTS Calling Party address handling

This release adds the option to select the calling party address in a UDTS/XUDTS to be either derived from the called party address of the original UDT/XUDT or the address of the local system. By default the module uses the local pc and an unknown SSN for the calling party address of the UDTS/XUDTS. If the option SCPXF_SWAP_UDT_ADDR (0x2000) is set then the called and calling party addresses in the UDT/XUDT are swapped for the UDTS/XUDTS. If required the option should be set in the extended options field of the SCP_MSG_CONFIG (0x7740) message. The option has no affect on the addresses which are not in UDTS/XUDTS messages.

# 3 Other Changes

## 3.1 Correction of GTT request SCP_MSG_R_GTT

In the previous software release the odd/even bit (encoding scheme) returned in SCP_MSG_R_GTT may be wrong when the GTT table contained more than one rule. This problem has now been resolved and the software will always return the correct odd/even bit value.

## 3.2 UDTS/XUDTS generation on SCCP link congestion

Previously if a message could not be routed due to network congestion an SCCP Maintenance event with reason SCPEV_RTF_NET_CONG (4) was generated but no UDTS/XUDTS was set back to the sending node. This release now sends back an UDTS or XUDTS as appropriate.

Dialogic
26-Jun-06

**Binary for Windows - SCCP**

**Release Notes for Version 5.01**

# 1 Overview

This release introduces Global Title Load Share Tables (GLST) for the load sharing of messages matching a specific global title translation rule across several destination point codes. GLST allows the user to select one of two forms of routing based resource availability. The two modes are appropriate either for routing via a number of intermediate signalling points or for routing direct to a number of end points.

This release extends the number of sub-system resources (SSR) which can be configured to 512 and the number of sub-system resource entities that can be concerned about a particular sub-system resource (SSR) to 32. This release introduces enhanced functionality to allow systems making use of the SMB broadcast functionality to offer faster update of RSS availability on restart of a SCCP module. The release also includes a number of other changes to add new functionality and clear faults as detailed below.

This release fixes a fault that prevented some SCCP Connection Orientated messages being sent to MTP3.

Customers using message segmentation, Connection Oriented operation or requiring any of the new functionality should upgrade to this release. Other users need not upgrade.

# 2 New functionality

## 2.1 Global Title Load Share Tables

Global Title Load Share Tables support the distribution of messages for a configured Global Title Translation Rule across a number of point codes. A Load Share Table holds a number of Point Codes across which messages will be distributed. The Point Code selection is made using the SLS value for messages received from the network, or Sequence Control parameter for messages received from the User Part. This supports in sequence delivery of messages for a particular TCAP dialogue.

The configuration requires three additional message types (to add, modify or remove GLST tables) and the addition of two optional parameters to the existing GTT configuration message (SCP_MSG_GTT_ADD).

**In sequence delivery and loadsharing.**
For TCAP messages both the SLS and seq_ctrl parameter are derived

from the TCAP transaction id such that all messages with the same transaction id (and thus requiring in sequence delivery) have the same SLS and seq_ctrl value.

In the absence of network failure or sub-system management events, any message routed using GLSTs with the same MTP SLS or SCCP SEQUENCE_CONTROL value will be sent to the same destination point code (DPC). The DPC to be used for a particular message is determined by indexing a GLST table with either the messages SLS value for messages received from the network, or the seq_ctrl parameter for messages received from the user.

In the event of a point code in a GLST associated with a particular SLS or seq_ctrl value to become unavailable, the next entry in the GLST table will be used. If the next entry is also unavailable then the next will be considered until all DPCs have been considered. If no DPC is found from the GLST table then the message will be routed to the Point Code found from the GTT translation itself.

**Operation Modes**
Two modes of GLST operation are supported:

- Load sharing via a number of intermediate signalling points

- Load sharing direct to a number of end points

The GLST operation modes are selected using the SCP_GLSTF_PC_AND_SS_AVAIL option bit of a new ‘options’ parameter defined for the GLST Add and Modify messages.

The option bit selects the following modes of operation :

- SCP_GLSTF_PC_AND_SS_AVAIL = 0 : this setting should be used for load sharing via a number of intermediate signalling points.
  In this mode, if the point code from Primary GTT translation is available, the Primary GLST table is checked. If the point code from Primary GTT translation is unavailable, the Backup GTT translation used and the Backup GLST table is checked.
  When checking the GLST tables, the table entry is used if the module knows that the entry’s point code is available.

- SCP_GLSTF_PC_AND_SS_AVAIL = 1 : this setting should be used for load sharing direct to a number of end points.
  In this mode, after the Primary GTT translation is performed the Primary GLST table is checked. If the Primary GLST (or Primary GTT) point code resulting from this check cannot be used (see below), the Backup GTT translation is performed and the Backup GLST table is checked.
  When checking the GLST tables, the table entry is used if the module knows that both the point code is available <u>and</u> the sub-

system ‘has not been reported as being unavailable’. This means that if the point code is available, the entry will be used if: the sub-system has been reported as being available; or the sub-system has not been configured; or the translation being tested has no sub-system number.

It is recommended that the SCP_GLSTF_PC_AND_SS_AVAIL option bit is set to the same value for both Primary and Backup GLST tables. Where the tables to have different settings, the Primary GLST option bit would select when the GLST tables are checked (as described above) and the availability test used for the Primary GLST table entries. The Backup GLST option bit would only define the availability test used for the Backup GLST table entries.

**Called Party Point Code**
The Point Code is normally not left in the SCCP called party address when the message is sent to the network as it is also in the MTP3 Routing Label. If the Point Code is to be left in the message then the Point Code from the Load Share Table will be used.

**Use of Local Point Code in GLST tables**
GLST tables can be configured to include the Local Point Code (LPC). When this is the case, messages are directed to the LPC according to the GLST load sharing rules with the exception that the addressed local sub-system (LSS) must be available for the LPC to be used, i.e. the LSS is tested for availability even if the SCP_GLSTF_PC_AND_SS_AVAIL option bit is set to 0. The LPC is always considered to be available.

The following messages are added to support GLST operation.

### 2.1.1 Add Global Title Load Share Table Request

**Synopsis:**
Message used to add a Global Title Load Share Table for reference by a Global Title.

**Message Format:**

| MESSAGE HEADER | | |
|---|---|---|
| **FIELD NAME** | | **MEANING** |
| **type** | | **SCP_MSG_GLST_ADD** (0x774E) |
| **id** | | GLST id. |
| **src** | | Sending module_id |
| **dst** | | **SCP_TASK_ID** |
| **rsp_req** | | used to request a confirmation |
| **hclass** | | 0 |
| **status** | | 0 |
| **err_info** | | 0 |
| **len** | | 136 |
| **PARAMETER AREA** | | |
| **OFFSET** | **SIZE** | **NAME** |
| 0 | 4 | **options** – option flags |
| 4 | 2 | **reserved** – set to 0 |
| 6 | 2 | **num_pc** |
| 8 | 128 | **dpc[ num_pc ]** – Array of destination point codes. |

**Description:**

Load Share Tables may be added to the system at any time providing that the maximum number supported by the system is not exceeded.

**id**

Logical id of the GLST instance.

**options**

Option flags to control GLST functionality for this GLST instance. The bit mapped options are shown in the following table -

| Bit | Mnemonic | Description |
|---|---|---|
| 0 | SCP_GLSTF_PC_AND_SS_AVAIL | If set, the point code and the sub-system is tested for availability. If not set, only the point code is tested for availability. |
| 1-31 | | All other bits are reserved for future use and must be set to zero. |

**num_pc**

The number of destination point codes in the following array.

**dpc[ ]**

Array of up to 32 * 4 byte point codes across which messages will be load shared. If less than 32 point codes are submitted the user should pad the end of the array with zero to the correct length.

### 2.1.2 Modify Global Title Load Share Table Request

**Synopsis:**

Message used to modify a previously configured Global Title Load Share Table.

**Message Format:**

| MESSAGE HEADER | | |
|---|---|---|
| **FIELD NAME** | | **MEANING** |
| **type** | | **SCP_MSG_GLST_MOD** (0x774F) |
| **id** | | GLST id. |
| **src** | | Sending module_id |
| **dst** | | **SCP_TASK_ID** |
| **rsp_req** | | used to request a confirmation |
| **hclass** | | 0 |
| **status** | | 0 |
| **err_info** | | 0 |
| **len** | | 136 |
| **PARAMETER AREA** | | |
| **OFFSET** | **SIZE** | **NAME** |
| 0 | 4 | **options** – option flags |
| 4 | 2 | **reserved** – set to 0 |
| 6 | 2 | **num_pc** |
| 8 | 128 | **dpc[ num_pc ]** – Array of destination point codes. |

**Description:**

Load Share Tables may be modified at any time.

**id**

Logical id of the GLST instance.

**options**

Option flags to control GLST functionality for this GLST instance. The bit mapped options are shown in the table for section 2.1.1.

**num_pc**

The number of destination point codes in the following array.

**dpc[ ]**

Array of up to 32 * 4 byte point codes across which messages will be load shared. If less than 32 point codes are submitted the user should pad the end of the array with zero to the correct length.

### 2.1.3 End Global Title Load Share Table Request

**Synopsis:**

Message used to end a previously configured Global Title Load Share Table.

**Message Format:**

| MESSAGE HEADER | | |
|---|---|---|
| **FIELD NAME** | | **MEANING** |
| **type** | | **SCP_MSG_ GLST_REM** (0x7750) |
| **id** | | GLST id. |
| **src** | | Sending module_id |
| **dst** | | **SCP_TASK_ID** |
| **rsp_req** | | used to request a confirmation |
| **hclass** | | 0 |
| **status** | | 0 |
| **err_info** | | 0 |
| **len** | | 0 |
| **PARAMETER AREA** | | |
| **OFFSET** | **SIZE** | **NAME** |

**Description:**

Load Share Tables may only be ended if no GTT rules are currently referencing it.

**id**

Logical id of the GLST instance.

**New GTT Configuration Parameters**

The SCP_MSG_GTT_ADD and SCP_MSG_GTT_MOD commands now support the following parameters to identify GLST tables for use.

| | |
|---|---|
| Parameter name | SCPPN_PRIMARY_GLST (value 0x20) |
| Parameter length | Fixed to 2 |
| Parameter data | Logical ID of the GT Load Share Table for use with the Primary translation. |

| | |
|---|---|
| Parameter name | SCPPN_BACKUP_GLST (value 0x21) |
| Parameter length | Fixed to 2 |
| Parameter data | Logical ID of the GT Load Share Table for use with the Backup translation. |

**Change to use of the SCP_MSG_GTT_MOD message type**
The SCP_MSG_GTT_MOD (0x7747) message will now accept any parameter in addition to the reference *Target address*. The user can supply the *Primary address, Backup address, GTT Mask, Primary GLST ID* or the *Backup GLST ID* or any combination of these. Previously the message had to supply at least the *Primary address* and/or the *Backup address*. For both the *Primary address* and the *Backup address*, if they are not supplied any existing configuration for that parameter is retained. For the other parameters (*GTT Mask, Primary GLST ID* or *Backup GLST ID*), if they are not supplied any existing configuration for that parameter is removed.

## 2.2 Message gapping for Concerned LSS upon PAUSE

When MTP-Pause or MTP-Resume is received for a Remote Signalling Point (RSP), all local sub-systems, that have been set up as concerned, will be informed of the status change. The gapping algorithm used to limit the maximum rate of messages has been enhanced to better support systems with large numbers of resources.

## 2.3 RSS Availability Synchronization

Where SCCP is configured to use an SMB broadcast mechanism this release includes functionality to enable Remote Sub-Systems (RSS) to become available to a restarting SCCP module faster.

In the case where an SCCP module is started or restarted on reception of an MTP Resume for a remote point code which matches that of remote subsystems a synchronization procedure is triggered. This is optionally enabled using the SMB flag value 0x1000 in the main SCCP configuration message. The procedure causes a message SCP_MSG_SMB_SSR (0x774d) to be sent to the SMB broadcast module id. This message should be forwarded by the broadcast module to all active SCCP modules.

When the SCP_MSG_SMB_SSR message is received by an SCCP module if it believes the RSS is currently active and this was a result of a previous SSA message then an SSA will be broadcast to the SMB module id and hence on to all other SCCP modules in the system. This will cause the RSS to be marked as available without the restarting SCCP module to need to wait for the sub-system test procedure to be completed.

*Note: If the RSS is not known to be active, no message is sent as a reply. If the RSS is believed to be active but this has not been verified by the reception of an SSA then again no message reply is sent. These measures ensure that at least one valid and successful sub-system test procedure has been followed before any SSA is sent.*

*Note: If the If the SSRF_RSS_AVAIL SSR configuration flag is set then the synchronization is not performed as the RSS is marked as available immediately.*

The message format for SCP_MSG_SMB_SSR is included below but does not need to be sent by the user.

### 2.3.1 Read SCCP SSR Statistics Request

**Synopsis:**

Message used to request an SCCP module to broadcast a state indication for the specified SSR.

**Message Format:**

| **MESSAGE HEADER** | | |
|---|---|---|
| **FIELD NAME** | | **MEANING** |
| **type** | | **SCP_MSG_SMB_SSR_REQ** (0x774d) |
| **id** | | 0 |
| **src** | | **SCP_TASK_ID** |
| **dst** | | **SCCP SMB TASK ID** |
| **rsp_req** | | 0 |
| **hclass** | | 0 |
| **status** | | 0 |
| **err_info** | | 0 |
| **len** | | 9 |
| **PARAMETER AREA** | | |
| **OFFSET** | **SIZE** | **NAME** |
| 0 | 1 | **version** |
| 1 | 1 | **ssr_type** |
| 2 | 2 | **nc** |
| 4 | 4 | **spc** |
| 8 | 1 | **ssn** |

**Parameters:**

**version**
Must be set to 0.

**ssr_type**
As defined for **SCP_MSG_CNF_SSR** message.

**nc**
Network Context
**spc**
The signalling point code of the resource.

**ssn**
The sub-system number of the resource.

## 2.4 Modified GTT handling for GTI, ES and NoA

During Global Title Translation previous releases did not consider the GTI, ES and NoA parameters in the GT to be significant for the translation selection. This release now compares these parameters, where present, in order to select the correct translation. If this behavior is not desired then the SCCP extended option SCPXF_DIS_GTI can be used to disable this.

## 2.5 Increased sub-system resources

This release extends the number of sub-system resources (SSR) which can be configured to 512. These resources are shared between local sub-systems, remote sub-systems and remote point codes.

## 2.6 Increased concerned entity support

This release increased the number of sub-system resource entities that can be concerned about a particular sub-system resources (SSR). It is now possible for up-to 32 other SSR to be concerned about a specified SSR.

## 2.7 Allow CO messages with no data field to be sent

In previous releases a fault prevented messages with no SCCP data field being sent to MTP3. This only affected SCCP Connection Orientated (CO) messages, since SCCP Connection Less messages always have a data field.

The fault is fixed in this release. The module no longer rejects CO messages without a data field. Users not using Connection Orientated messages would not be affected by this fault.

## 2.8 Add field for trace module id to configuration message

A Trace Module ID can now be defined in the SCCP Module Configuration message (SCP_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 SCCP Module Configuration Message is now as follows:

| **MESSAGE HEADER** | |
|---|---|
| **FIELD NAME** | **MEANING** |
| **type** | **SCP_MSG_CONFIG** (0x7740) |
| **id** | 0 |
| **src** | Sending module_id |
| **dst** | **SCP_TASK_ID** |
| **rsp_req** | used to request a confirmation |
| **hclass** | 0 |
| **status** | 0 |
| **err_info** | 0 |
| **len** | 60 |

| **PARAMETER AREA** | | |
|---|---|---|
| **OFFSET** | **SIZE** | **NAME** |
| 0 | 1 | **cnf_ver** - must be set to one |
| 1 | 1 | **sio** - Service Information Octet value |
| 2 | 2 | **options** |
| 4 | 1 | **module_id** - SCCP module id |
| 5 | 1 | **mtp_id** - MTP module id |
| 6 | 1 | **mngt_id** - Management module id |
| 7 | 1 | **maint_id** - Maintenance module id |
| 8 | 4 | **point_code** - Local point code |
| 12 | 2 | **max_sif** |
| 14 | 1 | **sccp_instance** |
| 15 | 1 | **smb_id** - Broadcast module id |
| 16 | 2 | **smb_flags** - Broadcast flags |
| 18 | 2 | **num_uc** - Required number of User connections |
| 20 | 2 | **uc_onset** - Onset of congestion point for user connections |
| 22 | 2 | **uc_abmt** |
| 24 | 2 | **num_ic** |
| 26 | 2 | **ic_onset** |
| 28 | 2 | **ic_abmt** |
| 30 | 2 | **num_data** |
| 32 | 2 | **data_onset** |
| 34 | 2 | **data_abmt** |
| 36 | 2 | **num_edata** |

| MESSAGE HEADER | | |
|---|---|---|
| FIELD NAME | | MEANING |
| 38 | 2 | **edata_onset** |
| 40 | 2 | **edata_abmt** |
| 42 | 2 | **ext_options** |
| 44 | 2 | **base_id** |
| 46 | 2 | **top_id** |
| 48 | 2 | **min_id** |
| 50 | 2 | **max_id** |
| 52 | 1 | **isup_id** |
| 53 | 2 | **nc** |
| 55 | 1 | **hop_counter** |
| 56 | 2 | **error_offset** |
| 58 | 1 | **trace_id** – Trace module id |
| 59 | 1 | **reserved – set to zero** |

**trace_id**

Module identifier defining the destination for all trace messages.

# 3 Other Changes

## 3.1 SCP_MSG_END_CONC message fault

The previous release included a fault that occasionally caused the SCP_MSG_END_CONC message to invalidly remove another concerned resource in addition to that in the command. The fault only occurred after several concerned resources had been added and the resource to be removed in the command was not the last added.

The fault caused the resource erroneously removed to stop getting notification messages. The SCP_MSG_END_CONC message itself would be rejected with an error status if it attempted at a later stage to remove the resource. This fault is now cleared in this release.

## 3.2 SCP_GTT_MOD message bug

A problem had been found when using the SCP_GTT_MOD message. The module can sometimes reject valid (previously successful) messages with status code 6. This release corrects this fault. Valid SCP_GTT_MOD messages should no longer be rejected.

## 3.3 Failure to send to more than 8 Concerned RSP

The previous release would only inform the first 8 (out of a maximum 32) remote signalling points (RSP) set up as concerned resources for a

local sub-system. This has been corrected so that all 32 RSPs will be notified.

### 3.4 Segmentation fails when messages mix

A fault in the SCCP module has been fixed in the reassembly of segmented messages received from the MTP3 module. The fault occurred when the first segment of a message is received followed by all segments of a second message. The remaining segments of the first message are then rejected by the module when they are received. When this happens a message is not released back to the message pool. If segmented messages are continually being received in this way, the message pool will be exhausted.

The fault is fixed in this release. The module no longer rejects segmented messages when they are mixed together and all messages allocated by the module are released correctly.

Customers not using segmentation and reassembly procedures would not be affected by this fault.

### 3.5 Update Count of Local Concerned Resources

A fault has been fixed that could have affected the gapping of status change messages sent to local sub-systems. These messages should be sent in bursts of eight with a gap between bursts. The fault could cause less messages to be sent in a burst. The fault was only possible when local concerned resources have been added then removed using the SCP_MSG_ADD_CONC and SCP_MSG_END_CONC messages.

### 3.6 Allow CO messages with no data field to be sent

In previous releases a fault prevented messages with no SCCP data field being sent to MTP3. This only affected SCCP Connection Orientated (CO) messages, since SCCP Connection Less messages always have a data field.

The fault is fixed in this release. The module no longer rejects CO messages without a data field. Customers not using Connection Orientated messages would not be affected by this fault.

Dialogic
05-Mar-09

**Binary for Windows - SCCP**

**Release Notes for Version 5.03**

# 1 Overview

This release allows the user to send Unitdata Service (UDTS) and Extended Unitdata Service (XUDTS) messages on demand. The user can use this feature to re-transmit received UDTS/XUDTS messages. The existing automatic generation of UDTS/XUDTS within SCCP is sufficient for most users in standard applications.

This release also extends the existing Global Title Load Share Tables which supported Class 1 messages so that the functionality also applies with Class 0 messages. The release also includes a number of other changes as detailed below.

In this release 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. This is the first generally available release since V5.01

# 2 New functionality

## 2.1 User generation of SCCP UDTS/XUDTS

In most situations where a UDTS/XUDTS message is required to be sent, it is automatically generated by the SCCP module. In some situations it may be desirable for a received UDTS/XUDTS to be passed up to the user in a Notice indication and for this to be used to generate an onward UDTS/XUDTS to a further node. In order for a user to do this the SCCP module now supports generation of UDTS and XUDTS messages on demand. A new primitive type octet is added for this purpose as shown in the table below.

| Primitive | Mnemonic | Value (Decimal) | Value (Hex) |
|---|---|---|---|
| N_UDTS request | SCPPT_N_UDTS_REQ | 23 | 0x17 |

The following table lists the parameters associated with the new service.

| Primitive | Parameters | Status |
|---|---|---|
| N-UDTS request | SCPPN_CALLED_ADDR | MANDATORY |
| | SCPPN_CALLING_ADDR | MANDATORY |
| | SCPPN_USER_DATA | MANDATORY |
| | SCPPN_REASON_FOR_RET | MANDATORY |
| | SCPPN_HOP_COUNTER | OPTIONAL |
| | SCPPN_SEQ_CTRL | OPTIONAL |
| | SCPPN_SEGMENTATION | OPTIONAL |
| | SCPPN_IMPORTANCE | OPTIONAL |

If the Hop Counter parameter is present, an Extended Unitdata Service Message (XUDTS) will be issued to the network. If this parameter is not present, a Unitdata Service message (UDTS) will be issued.

Since UDTS and XUDTS messages do not define a ‘Protocol Class’, the presence of the Sequence Control parameter only affects the SLS value sent to the network.

Receive indication messages for the primitive type N_NOTICE may now receive the additional parameters: SCPPN_SEGMENTATION, SCPPN_MTP_SIO_LABEL and SCPPN_IMPORTANCE. These parameters are added to allow user to re-create all fields from a received UDTS/XUDTS message if it is relayed. The new parameter table for the N_NOTICE service is defined below.

| Primitive | Parameters | Status |
|---|---|---|
| N-NOTICE indication | SCPPN_REASON_FOR_RET | MANDATORY |
| | SCPPN_CALLED_ADDR | MANDATORY |
| | SCPPN_CALLING_ADDR | MANDATORY |
| | SCPPN_USER_DATA | MANDATORY |
| | SCPPN_HOP_COUNTER | OPTIONAL |
| | SCPPN_SEGMENTATION | OPTIONAL |
| | SCPPN_MTP_SIO_LABEL | OPTIONAL |
| | SCPPN_IMPORTANCE | OPTIONAL |

In order to support the new messages a new ‘Importance’ parameter type is added. See the parameter table defined below.

| Parameter | Mnemonic | Value (Decimal) | Value (Hex) |
|---|---|---|---|
| ... | ... | ... | ... |
| Segmentation | SCPPN_SEGMENTATION | 29 | 0x1d |
| MTP SIO Label | SCPPN_MTP_SIO_LABEL | 30 | 0x1e |
| Network Context | SCPPN_NC | 31 | 0x1f |
| Primary GLST ID | SCPPN_PRIMARY_GLST | 32 | 0x20 |
| Backup GLST ID | SCPPN_BACKUP_GLST | 33 | 0x21 |
| Destination Transaction ID | SCPPN_DTID | 34 | 0x22 |
| Importance | SCPPN_IMPORTANCE | 35 | 0x23 |

| | |
|---|---|
| Parameter Name | SCPPN_IMPORTANCE |
| Parameter Length | Fixed , set to 1 |
| Parameter Data | A value between 0 and 7. Importance parameter encoded in accordance with Q.713. |

The SCPPN_IMPORTANCE parameter is currently only used with the SCPPT_N_UDTS_REQ and SCPPT_N_NOTICE_IND services. It is ignored for ANSI messages.

The SCPPN_SEGMENTATION parameter can now be defined for messages for transmission. Parameter data is inserted directly into the network message for the Segmentation field. Currently only the SCPPT_N_UDTS_REQ service accepts the parameter for transmission. Note: That the parameter is always accepted for transmission, e.g. it is not dependant on the SCPXF_SEGMENTATION or SCPXF_ONBEHALF_REASSEMBLY option flags.

| | |
|---|---|
| Parameter Name | SCPPN_SEGMENTATION |
| Parameter Length | Fixed , set to 4 |
| Parameter Data | Segmentation parameter encoded in accordance with Q.713. The first octet defines flags and the remaining segment count. The 3 following octets are the Segmentation Local Reference number. |

On reception the SCPPN_SEGMENTATION and SCPPN_MTP_SIO_LABEL parameters may be returned if the SCPXF_ONBEHALF_REASSEMBLY option flag is set or the message is XUDTS and the SCPXF_SEGMENTATION option flag is set. If neither option flag is set, a received segmented message will be rejected and a SCPEV_RTF_CANNOT_REASSEMBLY maintenance event reported as before.

| | |
|---|---|
| Parameter Name | SCPPN_MTP_SIO_LABEL |
| Parameter Length | Fixed , set to 5 |
| Parameter Data | Received message's SIO and Routing Label data as defined by Q.704. |

For UDTS/XUDTS messages, no automatic message segmentation or re-assembly is performed, i.e. the SCCP module ignores the SCPXF_SEGMENTATION option if set. Received segmented UDTS/XUDTS messages are returned to the user as individual messages with SCPPN_SEGMENTATION, SCPPN_MTP_SIO_LABEL and SCPPN_IMPORTANCE parameter data as appropriate.

*Note: That the SCPXF_FORCE_XUDT option has been extended to apply to UDTS messages as well as UDT messages. So that if the option is set, the module will automatically add a default Hop Counter parameter to a user UDTS request and send it as a XUDTS message.*

## 2.2 Global Title Load Share handling of Class 0 messages

Global Title Load Share Tables use the User SCPPN_SEQ_CTRL parameter (or received SLS value) to determine the Load Share Table entry (point code) to use. The SCPPN_SEQ_CTRL parameter is only present for Class 1 messages so the module has been enhanced so that for non-Class 1 messages, the Load Share Tables use the message's SLS value. The SLS value for these messages is an incrementing value, updated each time a (non-Class 1) message is sent.

Operation using Class 1 messages is not affected by this change.

*Note: It is recommended that Class 1 messages are used for Global Title Load Share Tables in order to preserve message sequence throughout a network whenever a dialogue may have more than one message sent in the same direction.*

## 2.3 SCCP Sub-System Resource initialisation state

The SSRF_RSSR_INIT option is added to the SCP_MSG_CNF_SSR message to define the initial availability state of the Sub-System Resource (SSR) assumed by the SCCP module. The option applies only to Remote Signalling Points (RSP) and Remote Subsystems (RSS), it is ignored by Local Sub-Systems (LSS). The initial availability states for the SSRs are defined in the table below:

| Sub-System | SSRF_RSSR_INIT option | |
|---|---|---|
| | 0 | 1 |
| **RSP** | Allowed | Prohibited |
| **RSS** | Allowed | Takes state from the RSP |
| **LSS** | Prohibited | Prohibited |

This option has been added to prevent the initial availability state of a SSR being out of step with the MTP3 state. This is especially applicable when a SSR configuration is added after the initial SCCP configuration.

### 2.4 max_sif configuration up to 315 octets

The max_sif parameter for the SCP_MSG_CONFIG message can now be set to a value up to 315 octets. This allows systems that use SCCP with non-MTP3 protocols such as M3UA, to send messages that are longer than 272 octets without needing to invoke the segmentation procedures. The user should ensure that the remote end is capable of receiving the messages generated if max_sif is set to a value greater than 272. Users using SCCP with MTP3 should still ensure that max_sif is configured to 272.

As this feature requires specific support by the remote end a more generically useful solution for most customers is to make use of SCCP Segmentation and Reassembly procedures.

## 3 Other Changes

### 3.1 SCP_MSG_R_SSR_STATS bss_msg count for LSS only

The ‘bss_msg’ count reported by the SCP_MSG_R_SSR_STATS message has been changed so that only Local Sub-Systems (LSS) will report a count. This is consistent with the Q.752 standard (section 3.4.5). This count is number of messages for a LSS that were directed to a backup sub-system due to the LSS being unavailable.

### 3.2 SCPF_PLIST option no longer supported

The SCPF_PLIST option for the SCP_MSG_CONFIG message is no longer supported. The ITU formatting associated with this option is no longer required and has been removed.

### 3.3 Support for dual license tokens

This release supports both the old style "INTEL" tokens and the new style "dialogic" tokens within the license file allowing it to function with

existing deployed licenses and support future enhancements. It is the first release of this binary to support the Dialogic named license token. This change should be of no consequence to the user.

### 3.4 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