Binary for Solaris - SCCP

Release Notes for Version 4.01

1. Overview

This release is the first release since V3.01 and includes enhancements to the licensing mechanism to support new format tokens in license files.

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

Customers using Connection Orientated messages should upgrade to this release. Other users need not upgrade. This release is backwards compatible with the previous release.

2. Fault cleared

2.1 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.

2.2 CO messages with no data field

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 corrected in this release.

Users of Connectionless Operation would not have been affected by this fault.

Dialogic
14-Nov-08

Binary for Solaris - SCCP

Release Notes for Version 4.03

1. Overview

This release extends the existing Global Title Load Share Tables which supported Class 1 messages so that the functionality also applies with Class 0 messages. This release also adds the ability to send trace messages to a nominated Trace module and corrects a number of faults as detailed below.

Other functionality in this release is the ability for 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.

It is the first release since V4.01 and is backwards compatible with that release.

2. New functionality

2.1 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.

2.2 User configurable trace module id

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 defined as –

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**
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.

2.3 User generation of SCCP UDTS/XUDTS messages

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.

3. Faults cleared

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 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.

3.2 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, e.g. M3UA, to send messages that are greater than the 272 octet limit imposed by MTP3. SCCP can generate valid non-segmented messages that are greater than 272 octets, setting a max_sif limit of up-to 315 octets allows these messages to be passed. The SCP_MSG_CONFIG message will be rejected if a max_sif value greater than 315 is set. 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.

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. SCCP Programmer’s Manual states that this option must be set to 1.