Binary for Solaris – INAP

Release Notes for Version 3.00

1. Overview

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

This is the first release since V2.06. 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 customer should upgrade to SCCP for Solaris V3.00 or later and TCAP for Solaris V4.00 or later. Other customers need not upgrade.

The INAP API is now delivered as a Linux shared object, giving the user the potential in the future to upgrade the library without needing to re-link the application process.

When using new INAP API shared object, it is necessary to compile with the IN_LMSGS #define to get the correct function prototypes described in section 2.3.2 below.

2. New functionality

2.1 Support for Long Messages in INAP module

Starting with this release the INAP module now supports the generation and reception of long TCPPN_COMPONENT (towards and from the TCAP) and long INAPPN_COMPONENT (towards and from the application). This is typically used to support SCCP Segmentation and Reassembly procedures together with the SCCP and TCAP modules.

To enable correct operation the INAPF_SEGMENTATION (0x0010) flag of the option field of INAP_MSG_CONFIG (0x77f4) must be set and the GCT environment must be set up to handle long messages.

See ga237sss.pdf for more details on the new encoding scheme allowing supporting parameters longer than 255 bytes in the INAP to TCAP interface.

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

As well as making use of the GCT shared object, the INAP API is also available as a shared object. See the sections below on the INAP API for further details.

2.3 Support for Long Messages in INAPAPI

2.3.1 INAP API changes

To enable the INAP API library to effectively make use of longer parameters and messages it was necessary for a new version of the library to be created with a slightly modified interface.

The new INAP API library is also implemented as a shared object in order to allow applications to make use of updated versions of the library without recompilation. For backwards compatibility, the old INAPAPI is still available as a statically linked library without support for Long Messages.

2.3.2. Functions Prototypes

```
int IN_init_component(void *prot_spec, IN_CPT *cpt, u32 options);

HDR *IN_alloc_message(u32 options);

int IN_set_dialogue_param(u16 param, u16 len, u8 *dptr, HDR *h);

int IN_get_dialogue_param(u16 param, u16 *dlen, u8 *dptr, u16
max_len, HDR *h);

int IN_set_component_param (u16 param_name, u16 len, u8
*data_ptr, IN_CPT *cpt);

int IN_set_param_length_range(void *prot_spec, u16 param_name,
u16 length_min, u16 length_max);

int IN_get_component_param(u16 param_name, u16 *data_len, u8
*data_ptr, u16 max_len, IN_CPT *cpt);
```

Function	Change	Notes
IN_init_component	Addition of options field	Bit 0 (IN_INIT_OPTION_CODE_SHIFT) set to indicate if Code Shift is supported (may be generated to INAP)
IN_alloc_message	Addition of options field	Bit 0 (IN_ALLOC_OPTION_LMSGS) set to indicate if the allocated message is a large message or not.
IN_set_dialogue_param		Parameter len now u16
IN_get_dialogue_param		Parameter dlen now pointer to u16 Parameter max_len now u16
IN_set_component_param		Parameter len now u16
IN_set_param_length_range		Parameters length_min, length_max now u16
IN_get_component_param		Parameter data_len now pointer to u16 Parameter max_len now u16

2.3.3 IN_CPT Structure Changes

It is recommended that no application modify the internals of the structure. If an application does so it should be noted that the IN_CPT structure itself, as well as the databuf structure in IN_CPT have been modified to support a longer parameter area.

2.3.2. Linking with the INAP API Shared Object

If the new shared object version of the INAP API library is used then the linker option which links in the library should be changed. For use with the INAP API library file called `libin_api.so` , the option `in_api` becomes `–lin_api`.

2.4 Additional error report in INAP_MSG_CONFIG confirmation

In order to simplify the identification of configuration errors INAP_MSG_CONFIG (0x77f4) now supports one extra field, error_offset (INAPMO_CONFIG_error_offset=24).

The module ignores this field on reception of the INAP_MSG_CONFIG message but sets the field in the INAP_MSG_CONFIG confirmation (0x37f4) 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 INAP_MSG_CONFIG which causes the configuration to fail. In order to do this the INAP_MSG_CONFIG message length must be long enough, i.e. at least 26 bytes. There is

no change of operation when INAP_MSG_CONFIG length is less than 26 bytes

3 Fault cleared

Ellipsis recovery

In previous software release, an ellipsis could only recover one “unknown” parameter: if an operation contained more than one “unknown” parameter, this may have resulted in a corrupted ellipsis.

In this software release, if more than one “unknown” parameter is present in an operation they are concatenated together in the order they appear in the message and stored in the same ellipsis parameter.

Message formatting to TCAP

In previous software release, if messages sent to INAP resulted in a TCAP component larger than 255 bytes, this may have caused an unexpected behaviour. This issue has been fixed. In this software release, an error is returned if the encoded TCAP component is larger than 255 bytes and the INAPF_SEGMENTATION option is not enabled. If the INAPF_SEGMENTATION option is set, an error is returned when the TCAP component is larger than 3000 bytes.

Dialogic
31-May-06

Binary for Solaris – INAP

Release Notes for Version 3.01

1. Overview

This routine release corrects a number of faults within the INAP module.

This release is backwards compatible with the previous release.

2. New functionality

2.1 Invalid Network Context Reporting

Previous releases would send the maintenance indication INAP_MSG_MAINT_IND (0x07f9) with the following reason in the case of an invalid context being received by the module:

Mnemonic	Value	id	Details
INAPME_INVALID_NC	2	nc	Message received with unrecognized Network Context

This is now replaced with the Software Event code below which is sent as part of the error indication INAP_MSG_ERROR_IND (0x07f8):

Mnemonic	Value	id	Details
INAPSWE_INVALID_NC	11	nc	Message received with unrecognized Network Context

3. Other changes

3.1 Range Check for Dialogues

Previously if base incoming or outgoing dialogue ids are not set to 0, and the number of dialogue ids configured is set to the maximum (for instance base_i=0x1000, base_o=0x9000, n_i=0x8000 and n_o=0x8000), this resulted in an error which was not handled by the INAP module.

An improved range check has now been implemented to correct this error.

3.2 Unexpected Linked Operation

The error code generated for an unexpected linked operation was incorrect. This is now fixed and will return the following now as part of a provider error:

Parameter name	INAPPN_provider_error
Parameter length	Fixed, set to one octet
Parameter data	6: Unexpected Linked Operation - The remote system was not expecting the linked operation invocation

3.3 INAP sends REJECT to TCAP after receiving END

If the module detected a syntax error in a component indication from TCAP, it would send a REJECT even if an END had been received. This would cause TCAP to assume that this REJECT belongs to another dialogue causing the abort of valid dialogues.

This issue has now been resolved and will not send an REJECT if an END has been received.

3.4 User AC not supported is not passed to TCAP

When the INAP module has the Transparent AC option flag set (INAPF_TRNS_AC) in the INAP configuration message INAP_MSG_CONFIG (0x77f4) to the module, open indications for all Application Contexts are passed to the user. The INAP user can abort the dialog and may give a user reason of 6 (documented as 'AC not supported'). When the module generates a TCAP U_ABORT message, the Abort Reason field was incorrectly set to 2 (Other Reason) and not 1 for 'AC not supported' as expected.

This fault has now been corrected and will return the correct reason.

Dialogic
15-Jul-08

Binary for Solaris – INAP

Release Notes for Version 4.01

1. Overview

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

This release is backwards compatible with the previous release.

2. Changes

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.

Dialogic
09-Apr-09