Binary for Solaris - SUA

Release Notes for Version 1.06

1. Overview

This is the first full release of the Dialogic® SUA Protocol Module which is intended for use in conjunction with the Dialogic® TCAP module and the SCTP modules contained in the Dialogic® SS7 Development Package for Solaris.

The implementation is based on the IETF RFC 3868 protocol specification and offers support for both ASP and IPSP operation. ASP operation is appropriate when connecting to SUA Signaling Gateways whilst IPSP operation permits connection to other IPSP nodes in a peer-to-peer configuration. Functionality in the modules includes Global Title Translation and Transaction Id based routing.

These release notes should be read in conjunction with the Dialogic® SUA Programmer's Manual, Issue 3.

All users should upgrade to this release in preference to earlier releases of the software that have been available during the field trial. This release includes support for a large numbers of global title translations as well as a number of other enhancements and fault resolutions.

2. New functionality

2.1 Global Title Translation Support

The SUA module incorporates an enhanced algorithm for Global Title Translation (GTT) which permits storage in excess of 1 million translations and supports both fixed and variable rules. Global Titles can be translated to a primary and a backup address.

GTT configuration uses the same API as the existing SCCP module. Global Titles can be configured either using the s7_mgt configuration utility or dynamically using the message-based interface as detailed in the Dialogic® SCCP Programmer's Manual.

Note: Where a zero value is configured for the number of GTT supported in the main SUA configuration message then 64K GTT are configured by default. This permits s7_mgt users to configure large numbers of GTT without including them in the config.txt file.

Note: Under certain conditions the maximum number of GTT supported by the SUA Module is restricted to 256 GTT rules. This is the case where specific digit(s) would be matched between a pair of variable

length wildcard parts. For fixed translations or those with a single variable length part the module supports over 1 million GTT

2.2 Capability Licensing

The SUA module is licensed for use through one of a number of purchasable licenses. Different licenses support different amount of traffic throughput and may also impose a limit on the number of SCTP associations supported as detailed in the following table.

Traffic throughput is calculated as the sum of transmitted data and received data as follows:

- Licensed Transmit Data (N-UNITDATA Request passed into SUA) includes Source and Destination addresses and the data parameter length + 13 Bytes for the MTP3 Routing Label and SCCP Headers.

- Licensed Receive Data (CLDT Received via SCTP) includes Source and Destination address parameter lengths divided by 2 + data parameter length + 13 Bytes as described above.

Note: The action of passing messages down through the SUA and SCTP protocol modules results in the addition of message headers. Consequently messages transmitted towards the network will contain less 'traffic' than the corresponding message when it is received from the network and passed up through the stack.

2.3 Long Message Support

When used in conjunction with the appropriate Dialogic® SCTP Module, SUA supports message lengths up to 4065 octets as defined by the ITU-T in Recommendation Q.715 for SCCP.

Note: When working in an SUA environment where message lengths including headers may exceed 320 octets it is essential that the user configures a pool of long messages when initialising the GCT environment. This is achieved using the NUM_LMSGS command in the system.txt file.

3. Faults Cleared

3.1 Routing based on TID cannot be specified by peer

The TCAP Transaction Id (TID) label is now passed correctly when the R_ASP goes active for a R_AS. This label gives the TCAP instance value and masking info and is saved within SUA for the R_ASP. When a message is received from the R_AS and a reply is generated by the L_AS, the DTID in the reply can be used with the TID info saved to identify the R_ASP to send the reply to.

Note: It is essential to use Binary for Solaris – TCAP V4.01 or later.

3.2 Traffic Mode cannot be specified by peer

Traffic mode can now determined by the peer by passing it in the ASP Active request. Local configuration of peer Load share is retained for use when connecting to a SG and as a default when connecting to a R_ASP.

3.3 Load sharing across multiple streams

Traffic will be load shared across the available streams (using the sequence control parameter), excluding stream 0 which is for management messages only – unless it is the only stream.

3.4 Support for 16-bit Point codes

This release supports the optional use of 16-bit Point code values.

3.5 GTT Error Checking

GTT previously differentiated Rules based on the Target digits, however if some of these digits were different but specified as ‘ignore’ in the GTT mask, the rule would be accepted. GTT now detects this and rejects the configuration Rule to avoid ambiguous translations.

For clarity GTT will now reject an Enhanced rule which does not have the same number of separators in the Target, Primary/Backup and GTT_MASK parameters, unless the Global Title Indicator for the Primary/Backup is set to “no global title included”.

Dialogic
15-May-08

Appendix: SUA Support in s7_mgt

1 Introduction

This appendix outlines the steps required to use the s7_mgt module to configure the Dialogic® SUA SIGTRAN protocol module.

This text is designed to be read in conjunction with the Dialogic® Programmer's Manual for SIGTRAN Host Software [1].

1.1 Related Documentation

[1] U04STN: Dialogic® Programmer's Manual for SIGTRAN Host Software.

[2] U07STN: Dialogic® SUA Programmer's Manual.

1.2 Terms and Abbreviations

The following abbreviations are used in this document:

DPC Destination Point Code

LAS Local application Server

MMI Man Machine Interface

RAS Remote Application Server

RSG Remote Signaling Gateway

SG Signaling Gateway

2 Configuration and Operation

2.1 Module Ids

S7_MGT assumes Protocols use specific Module Ids. It is important that these are configured correctly in the system.txt file. For the configuration of SUA systems the following modules IDs are assumed:

Protocol	Module ID
SCTPD	0xD0
SCTP	0xD1
S7_MGT	0xCF
MNGT	0xEF
SUA	0xC3

Protocol	Module ID
TCAP	0x14
MAP	0x15
IS41	0x25
INAP	0x35

3 PROTOCOL CONFIGURATION COMMANDS

This section describes the commands and parameters that are used in the protocol configuration file config.txt. These are used by the s7_mgt utility to perform single-shot configuration of the protocol stack at start-up.

3.1 SIGTRAN SUA Configuration Model

SUA SIGTRAN Hosts use the IETF SIGTRAN SUA protocol to carry SS7 SCTP traffic to and from a SG or, in IPSP communication, to a RAS.

IPSP operation

Signaling links to a Remote Application Server are instantiated using the SNSLI command. This gives details of the IP link connecting the Host to the RAS. This command also identifies the type of SS7 traffic to be used (e.g. ITU14). This command is used once for each Signaling link that is to be supported.

Each LAS is identified with a SNAPI command. A maximum of four LASs are supported. Each RAS must be identified with a SNRAI command. A maximum of 32 RASs are supported.

The SNLBI command is used to bind a LAS to a RAS or RSG. The SNALI command is used to bind an RAS to a specific link.

The SNRTI command is used to define a route. Each route is bound to a specific RAS or RSG by an SNRLI command. A maximum of 64 routes and route bind commands are supported.

The DPC must either be defined in the SNRAI command which defines the RAS or in any route which is subsequently bound to the RAS.

Signaling Gateway Operation

Links to the RSG are declared using the SNSLI command. This gives details of the IP link connecting the Host to the RSG and declares the presence of an RSG by including an SG parameter.

Each LAS is identified with a SNAPI command. A maximum of four LASs are supported. RSGs do not have to be explicitly declared – they are set up implicitly by the SNSLI command.

The SNLBI command is used to bind an LAS to an RSG. The SNALI command is not required as the RSG is bound to a specific link by the SNSLI command.

As in IPSP configurations, the SNRTI command is used to define a route and routes are bound to a specific RSG by an SNRLI command. A maximum of 64 routes and route bind commands are supported.

For SG connections, the DPC must be included in any routes which are bound to the RSG.

3.2 SIGTRAN Parameters

The Configuration of SUA uses the MMI format commands as described in [1].

The parameter set supported has been extended as follows:

Name	**Description / *Change***	**Range**	**Default**
SNTYPE	Type of SNLINK – *SUA type has been added*	M2PA, M3UA, SUA	M3UA
TID_START	Start bit for TCAP instance identifier	0 – 31	Disabled
TID_END	End bit for TCAP instance identifier	0 – 31	Disabled
TID_VALUE	Value of TCAP instance identifier	0 – 0xffff	Disabled
SSR_ID	Sub-system resource to be used by this LAS	0 – 511	0
SSN	Sub-System Number	0 – 255	0

3.3 Configuration Commands

The following commands will be used to configure the system:

SNSLI – SIGTRAN Signaling Link Initiation command

The command line to be used for SUA configurations is:

```
SNSLI:SNLINK=,IPADDR=,[SG=],[SNTYPE=],[SS7MD=],[IPADDR2=],
       [HPORT=],[PPORT=];
```

Example:

```
SNSLI:SNLINK=1,SNTYPE=SUA,IPADDR=10.10.10.2;
SNSLI:SNLINK=1,SNTYPE=SUA,SG=1,IPADDR=10.10.20.2;
```

If an SG parameter is present this indicates that this is a link to a RSG.

If no SG parameter is present then this is an IPSP link. For these links appropriate SNRAI and SNALI command are used to create the RASs and to attach them to the link.

SNAPI - SIGTRAN Application Server Initiate command

The command line to be used for SUA configurations is:

```
SNAPI:AS=,RC=,[SS7MD=[ITU14|ITU24|ANSI]],[SNTYPE=]
[TRMD=[OR|LS|BC]],[TID_START=],[TID_END=],[TID_VALUE=],
[SSR_ID=];
```

Most numerical identifiers used in SIGTRAN commands are numbered from 1 but, to provide compatibility with the SCCP_SSR command, values for the SSR_ID parameter start at 0.

If any TID parameter is present then all must be present

The OPC parameter is not supported for SUA configurations.

Example:

```
SNAPI:AS=1,RC=1,TID_START=11,TID_END=6,TID_VALUE=48;
```

SNRAI - SIGTRAN Remote AS Initiate command

The command line to be used for SUA configurations is:

```
SNRAI:RAS=,RC=,[DPC=],[SS7MD=],[NASP=];
```

If the DPC parameter is not present then it must be specified in all route configuration commands which are bound to this AS.

Example:

```
SNRAI:RAS=1,DPC=126;
```

SNLBI - SIGTRAN Local AS Binding Initiate command

This command binds a LAS to a RAS or RSG.

The command line to be used for SUA configurations is:

```
SNLBI:SNLB=,AS=,[RAS=],[SG=];
```

Either a RAS or a SG must be specified but not both.

The RAS or SG must already be bound to at least one Signaling link entity.

When this command is used for a SG the RSG entity is instantiated by this command – there is no separate command.

Example:

```
SNLBI:SNLB=1,AS=1,RAS=1;
SNLBI:SNLB=2,AS=2,SG=1;
```

SNALI - SIGTRAN AS Link Initiation

Command to bind a RAS entity to a specific link. This command is only used for IPSP links.

The command line to be used for SUA configurations is:

```
SNALI:[SNAL=],[RAS=],[SNLINK=];
```

Example:

```
SNALI:SNAL=2,RAS=1,SNLINK=1;
```

When a SNSLI command is used to create an SG link a dummy SNALI entry is created with SNAL identifier equal to the SNLINK identifier used. This SNAL identifier will not then be available for subsequent SNALI commands.

SNRTI - SIGTRAN Route Initiation Command

The command line to be used for SUA configurations is:

```
SNRTI:SNRT=,RC=,[DPC=],[AS=],[SNTYPE=],[SS7MD=],[NASP=],[SSN=];
```

Examples:

```
SNRTI:SNRT=1,SNTYPE=SUA,DPC=125,AS=1;
SNRTI:SNRT=2,SNTYPE=SUA,DPC=126;
```

If the LAS parameter is not specified then the route is attached to all LASs.

For SUA configurations routes to both RASs and RSGs may be configured.

SNRLI - SIGTRAN Route Link Initiation

The command line to be used for SUA configurations is:

```
SNRLI:SNRL=,SNRT=,[SG=],[RAS=];
```

Examples:

```
SNRLI:SNRL=1,SNRT=1,SG=1;
SNRLI:SNRL=2,SNRT=2,RAS=1;
```

Either an SG or an RAS parameter must be present but not both.

For SUA the command has been extended to allow routes to RASs to be bound.

4 Typical configuration scripts

4.1 Connection to a Remote SG

Configuration file to attach an SUA AS to a remote SG.

```
SCCP_SSR 0 LSS 0x08 0x2d 0 MAP

CNSYS:IPADDR=193.195.185.100;
SNSLI:SNLINK=1,SG=1,IPADDR=193.195.185.110,SNTYPE=SUA;
SNAPI:AS=1,RC=1,SNTYPE=SUA,SSR_ID=0;
SNLBI:SNLB=1,AS=1,SG=1;
SNRTI:SNRT=1,DPC=126,SNTYPE=SUA;
SNRLI:SNRL=1,SNRT=1,SG=1;

TCAP_CONFIG 0x0000 8192 0x8000 8192 0x0000 0
MAP_CONFIG 2
```

4.2 Connecting to a Remote AS

Configuration file to attach to a RAS using IPSP.

```
SCCP_SSR 0 LSS 0x08 0x2d 0 MAP

CNSYS:IPADDR=193.195.185.100;
SNSLI:SNLINK=1,IPADDR=193.195.185.110,SNTYPE=SUA;
SNAPI:AS=1,RC=1,SNTYPE=SUA,SSR_ID=0;
SNRAI:RAS=1,RC=101;
SNALI:SNAL=1,RAS=1,SNLINK=1;
SNLBI:SNLB=1,AS=1,RAS=1;
SNRTI:SNRT=1,DPC=201,AS=1,SSN=7,SNTYPE=SUA;
SNRLI:SNRL=1,SNRT=1,RAS=1;

TCAP_CONFIG 0x0000 8192 0x8000 8192 0x0000 0
MAP_CONFIG 2
```