Dialogic® DSI Diameter Stack

DMR Programmer's Manual

September 2013

www.dialogic.com

Copyright and Legal Notice

All contents of this document are furnished for informational use only and are subject to change without notice and do not represent a commitment on the part of Dialogic Inc. and its affiliates or subsidiaries ("Dialogic"). Reasonable effort is made to ensure the accuracy of the information contained in the document. However, Dialogic does not warrant the accuracy of this information and cannot accept responsibility for errors, inaccuracies or omissions that may be contained in this document.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH DIALOGIC® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN A SIGNED AGREEMENT BETWEEN YOU AND DIALOGIC, DIALOGIC ASSUMES NO LIABILITY WHATSOEVER, AND DIALOGIC DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF DIALOGIC PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHT OF A THIRD PARTY.

Dialogic products are not intended for use in certain safety-affecting situations. Please see http://www.dialogic.com/company/terms-of-use.aspx for more details.

Due to differing national regulations and approval requirements, certain Dialogic products may be suitable for use only in specific countries, and thus may not function properly in other countries. You are responsible for ensuring that your use of such products occurs only in the countries where such use is suitable. For information on specific products, contact Dialogic Inc. at the address indicated below or on the web at www.dialogic.com.

It is possible that the use or implementation of any one of the concepts, applications, or ideas described in this document, in marketing collateral produced by or on web pages maintained by Dialogic may infringe one or more patents or other intellectual property rights owned by third parties. Dialogic does not provide any intellectual property licenses with the sale of Dialogic products other than a license to use such product in accordance with intellectual property owned or validly licensed by Dialogic and no such licenses are provided except pursuant to a signed agreement with Dialogic. More detailed information about such intellectual property is available from Dialogic's legal department at 1504 McCarthy Boulevard, Milpitas, CA 95035-7405 USA. **Dialogic encourages all users of its products to procure all necessary intellectual property licenses required to implement any concepts or applications and does not condone or encourage any intellectual property infringement and disclaims any responsibility related thereto. These intellectual property licenses may differ from country to country and it is the responsibility of those who develop the concepts or applications to be aware of and comply with different national license requirements.**

Dialogic, Dialogic Pro, Dialogic Blue, Veraz, Brooktrout, Diva, Diva ISDN, Making Innovation Thrive, Video is the New Voice, VisionVideo, Diastar, Cantata, TruFax, SwitchKit, SnowShore, Eicon, Eiconcard, NMS Communications, NMS (stylized), SIPcontrol, Exnet, EXS, Vision, PowerMedia, PacketMedia, BorderNet, inCloud9, I-Gate, ControlSwitch, NaturalAccess, NaturalCallControl, NaturalConference, NaturalFax and Shiva, among others as well as related logos, are either registered trademarks or trademarks of Dialogic Inc. and its affiliates or subsidiaries. Dialogic's trademarks may be used publicly only with permission from Dialogic. Such permission may only be granted by Dialogic's legal department at 1504 McCarthy Boulevard, Milpitas, CA 95035-7405 USA. Any authorized use of Dialogic's trademarks will be subject to full respect of the trademark guidelines published by Dialogic from time to time and any use of Dialogic's trademarks requires proper acknowledgement.

The names of actual companies and products mentioned herein are the trademarks of their respective owners.

This document discusses one or more open source products, systems and/or releases. Dialogic is not responsible for your decision to use open source in connection with Dialogic products (including without limitation those referred to herein), nor is Dialogic responsible for any present or future effects such usage might have, including without limitation effects on your products, your business, or your intellectual property rights.

Publication Date: September 2013

Document Reference: U01DMR

Revision History

Issue	Date	Description
4	30-Sep-13	Addition of resource removal message definitions
3	22-Feb-13	Multiple updates including addition of new Notify Reason & Abort Reason parameter values.
2	04-Jan-13	Additional primitive types added to DMR Session Indications and Requests Additional statistics supported.
1	31-Oct-12	Initial Release for use during Dialogic® DSI Diameter Stack beta trial.

Note: The current version of this guide can be found at: http://www.dialogic.com/support/helpweb/signaling

Contents

Figures

1 Introduction

The Dialogic® DSI Diameter Stack is a software implementation of the IETF Diameter Base Protocol which is intended to facilitate development of user applications that interface to LTE and IMS networks for the implementation of services in the areas of: Mobility, Online Charging and Offline Charging.

The Dialogic® DSI Diameter Stack includes a message based binary Diameter Module, a Functional API Library and utility components and header files for use when developing a User Application.

Dialogic's Diameter Module (DMR) implements the Diameter Base Protocol offering a message based API to the User Application to control Diameter sessions. DMR is a member of the family of Dialogic® DSI Components and offers similar message-based interfaces and management capabilities to those offered for other SS7 and SIGTRAN protocol layers. DMR uses the services provided by the SCTP layer of the Dialogic® DSI SIGTRAN Stack for the transfer of messages between Diameter Peers.

Dialogic's Diameter Functional API is an API layer for use by the User Application allowing easy development of Diameter applications without being encumbered with complex encoding and decoding rules.

DMR is an event driven task that uses standard structured message types for communication with other layers of the protocol stack. These messages are used to convey the protocol primitives between DMR and the User application and between DMR and SCTP.

This manual provides details of the interface to DMR. It is intended for use during Configuration, Operation and Maintenance of systems that use the DSI Diameter Stack. Application Developers should refer instead to the Diameter Functional API Manual.

1.1 Applicability

This manual is applicable to the following software:

Dialogic® DSI Development Package for Solaris – Release 5.4.0 or later

Dialogic® DSI Development Package for Linux – Release 6.6.1 or later

1.2 Related Documentation

Current software and documentation supporting Dialogic® DSI components is available at: http://www.dialogic.com/support/helpweb/signaling

The following User Documentation relates to the use of the Dialogic® DSI Diameter Stack:

- *Dialogic® DSI Diameter Stack - Diameter Functional API Manual*
- *Dialogic® DSI Components - Software Environment Programmer's Manual*
- *Dialogic® DSI SIGTRAN Stack - SCTP Programmer's Manual*

2 General Description

2.1 Feature Overview

Features of the Dialogic® DSI Diameter Stack include:

- Implementation of Diameter Peer State Machine
- Transaction buffering and failover functionality
- Implementation of Session State Machines
- Implementation of Realm Routing Table
- Operation as Client or Server
- Functional API offering full encode/decode of messages
- Customizable API supporting interface extension and modification
- User loadable Diameter definition dictionaries

2.2 Module interfaces

The following figure (Figure 1) shows a Context Diagram for the Dialogic® DSI Diameter Stack - Diameter Module (DMR) showing the external interfaces to the module. Each interface is message based and connects to one of the following entities:

- Diameter User

  Typically using the Dialogic® DSI Diameter Stack Functional API libraries which provide access to the user module to allow it to send and receive the Diameter primitive messages.

- Management

  The module is configured using messages. Messages are also used to send indications of relevant events to the configured management module so they can be handled or logged.

- Timer

  The module derives timing signals from the timer module supplied as part of the Dialogic® DSI Development Package.

- SCTP

  The Dialogic® DSI SIGTRAN Stack SCTPN modules provide a message based interface and access to an SCTP network for message transport. This is the interface over which the Diameter payload messages are sent or received encoded as SCTP messages.

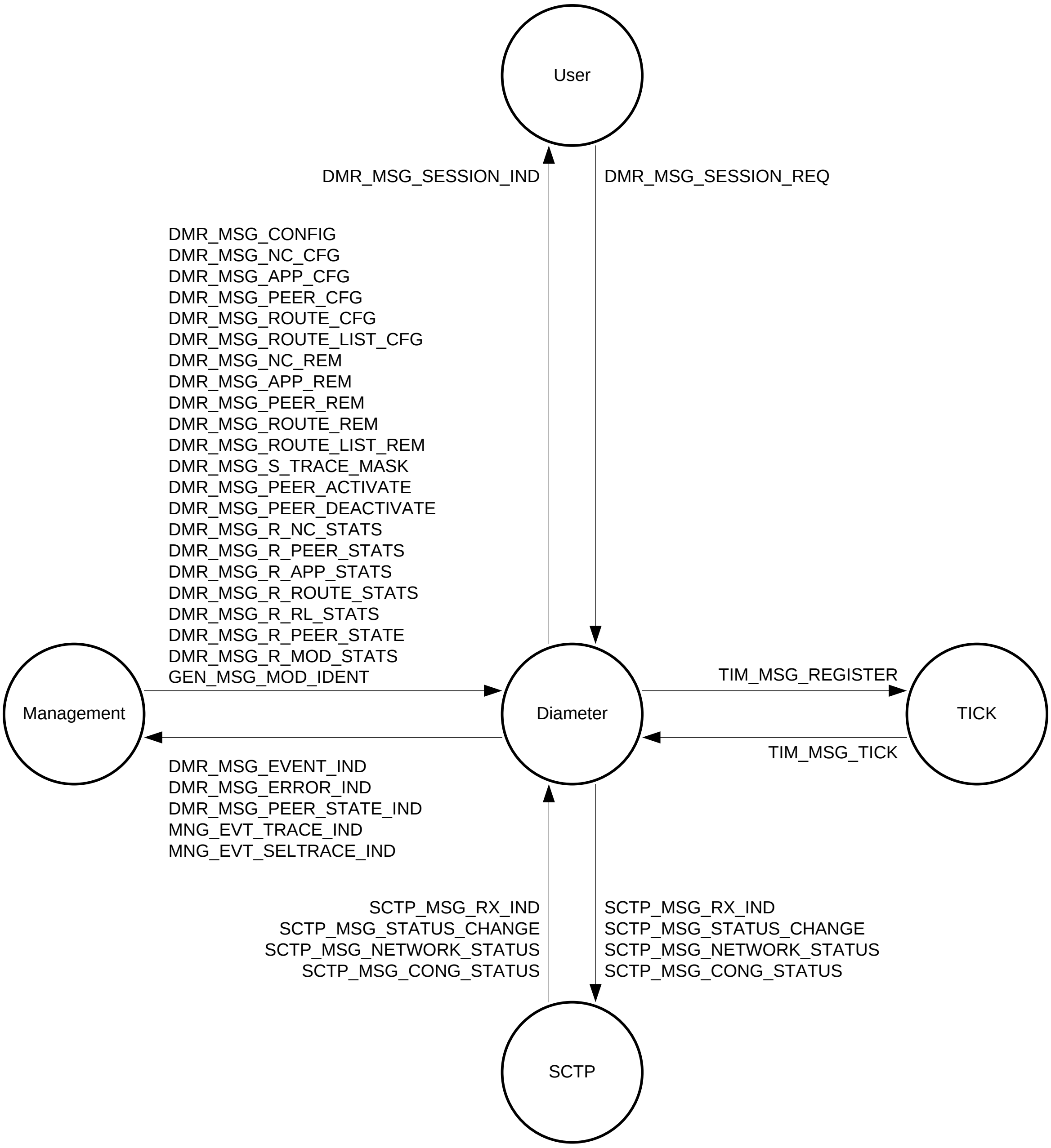


Figure 1. Dialogic® DSI Diameter Stack – DMR Context Diagram

2.3 Functional API

The Dialogic® DSI Diameter Stack is supplied with a supporting Diameter Functional API interface. This interface provides a mechanism to access all aspects of the message based interface exposed by the Diameter Module and is described further in the *Dialogic® DSI Diameter Stack - Diameter Functional API Manual*

2.4 Configuration

DMR is configured using messages which define the module set-up, including configuration and management settings to be used. The configuration model is discussed in further detail in section 3 and the messages used for configuration are defined in section 4.3. In most cases the user will not directly configure the module via these messages but will use the s7_mgt utility supplied in the DSI development package. It is also possible for the user application to generate the configuration messages directly.

2.5 Message Tracing

DMR provides tracing functionality to permit copies of messages sent or received by the module to be sent to a configurable trace module. This readily permits analysis and diagnosis of systems. The module can selectively trace different interfaces and message primitive types.

Further details of the message used to control the tracing can be found in section 4.4.3

2.6 Measurements

DMR supports a message based statistics gathering mechanism. A user application can send a message into the module for a configured Diameter Network context and the reply to the message will be populated with the relevant statistics such as session and transaction counts.

2.7 Event Reporting

DMR generates message based event and error indications to the configured management module for or logging protocol or operational events that have occurred.

Events generated include:

- Session Handling Event Indications
- Indications of errors detected during processing of messages
- Peer state changes

See section 4.5 Event Indication Messages for further details.

2.8 User Action

The Diameter protocol requires careful interpretation of answer messages in conjunction with runtime and configuration parameters such as DirectDebitFailureHandling or CreditControlFailureHandling to determine the correct action required by the Diameter Client. To aid compliance, interoperability and to ease application development the DMR module has been implemented to perform this interpretation for the client where appropriate. The client may then determine the required action by looking at the User Action parameter in the DMR_SESSION_IND message. This is defined in Section 4.2.3.

The User Action parameter is used in Credit Control applications (including the 3GPP Ro Online Charging interface) to indicate when service should be Granted or Denied, when messages should be Stored or Deleted and when a service error indication has occurred. It is used in Base Accounting applications (including the 3GPP Rf Offline Charging interface) to indicate when messages should be Stored or Deleted.

2.9 Policy Based Routing

DMR routing procedures use a routing table which is configured at system start time and used at run time. Route configuration and selection is made based on Diameter-Host or Diameter-Realm and Application-Id as per the Diameter specifications.

The Diameter module extends this functionality to offer greater routing control to the User application by allowing messages for a particular destination to be routed differently under the run time control of the user. For instance, the user application may wish to route all messages for a particular set of subscribers, or a particular command code via a different relay node for special processing. Or while testing a new feature certain messages may require routing via a test system.

This is achieved by the use of the optional ‘policy_id’ parameter when both configuring a route and sending DMR_SESSION_REQ messages. DMR maintains routing tables which contain the configured routing rules. All entries within a routing table must be distinct such that it is deterministic which route will be selected for a given Destination-Host, Destination-Realm and Application-Id. The policy_id allows the creation of multiple routing tables for the same Diameter Node. If a route configuration message has the policy_id parameter included, then the route will be considered part of that specific routing table. When a message is received by DSI Diameter for routing, if the policy_id is present then route selection will first search the corresponding route table.

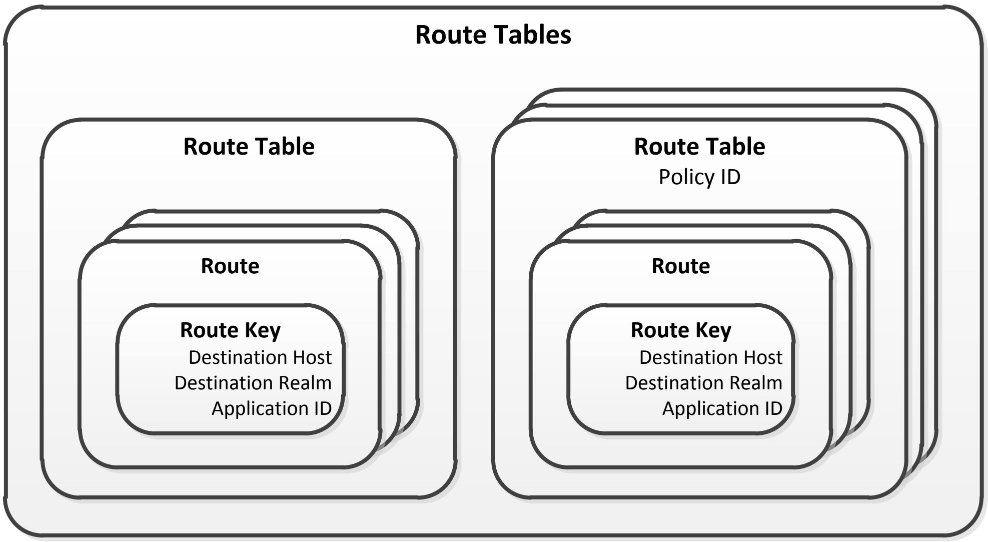


Figure 2: Route Table Structure

For example, if an application needs to route messages with Destination Realm 'Jupiter.com' via 'Peer Alpha' but wishes to route messages for certain subscribers via 'Peer Beta' in preference then this may be configured as follows:

1) The user configures Peer 0 (Peer Alpha)
2) The user configures Peer 1 (Peer Beta)
3) The user configures Route 0 with Destination Realm = 'Jupiter.com'. This route has no policy_id.
4) The user configures Route List 0 mapping Route 0 with Peer A
5) The user configures Route 1 with Destination Relam = 'Jupiter.com' and Policy_id = 1.
6) The user configures Route List 1 mapping Route 1 with Peer B (this will take the role of primary)
7) The user configures Route List 2 mapping Route 1 with Peer A (this will take the role of secondary)

This results in the configuration shown below.

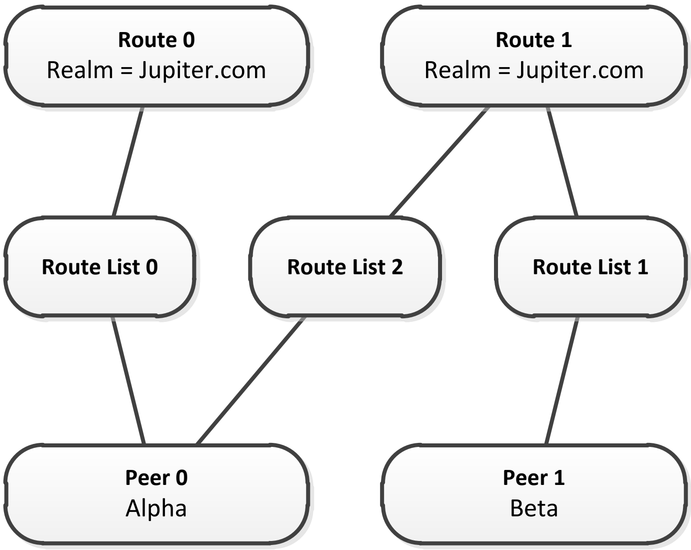


Figure 3: Example route configuration

At run time, the user application will determine which messages require special routing and will include the policy_id parameter set to the value 1. If the Destination-Realm equals 'Jupiter.com', then DMR will match and select Route 1 and, presuming it is available, Peer Beta.

The general flow through route selection logic is shown below.

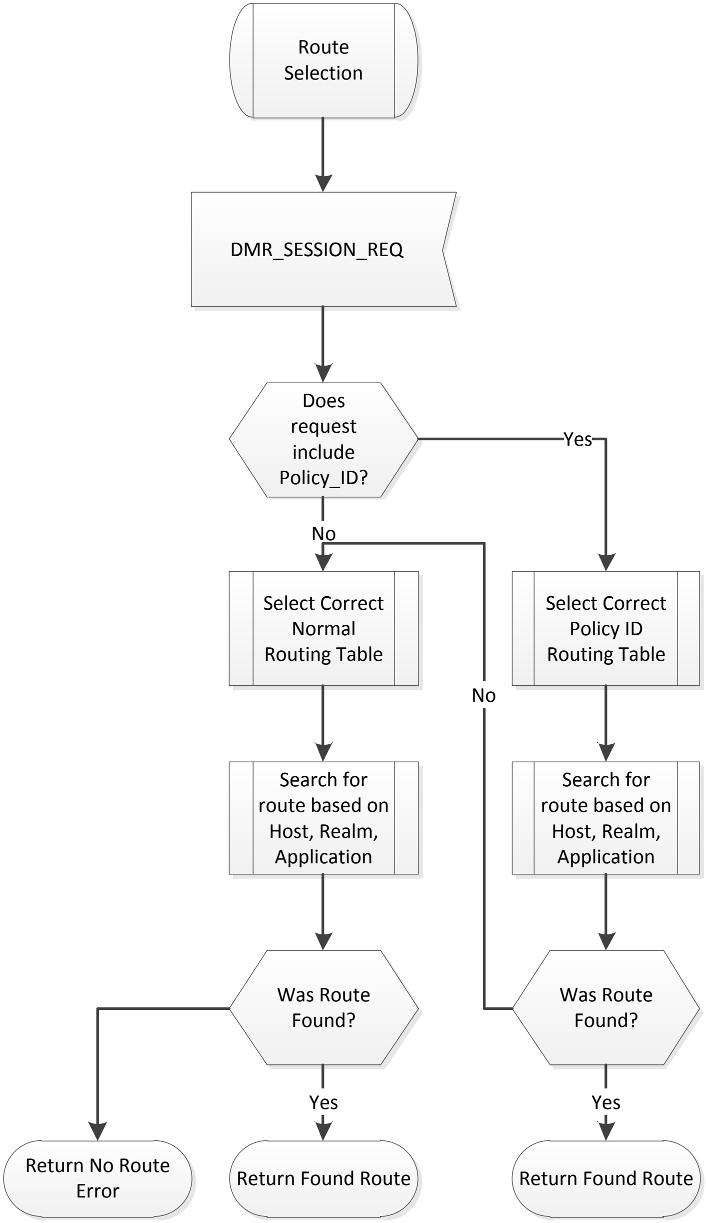
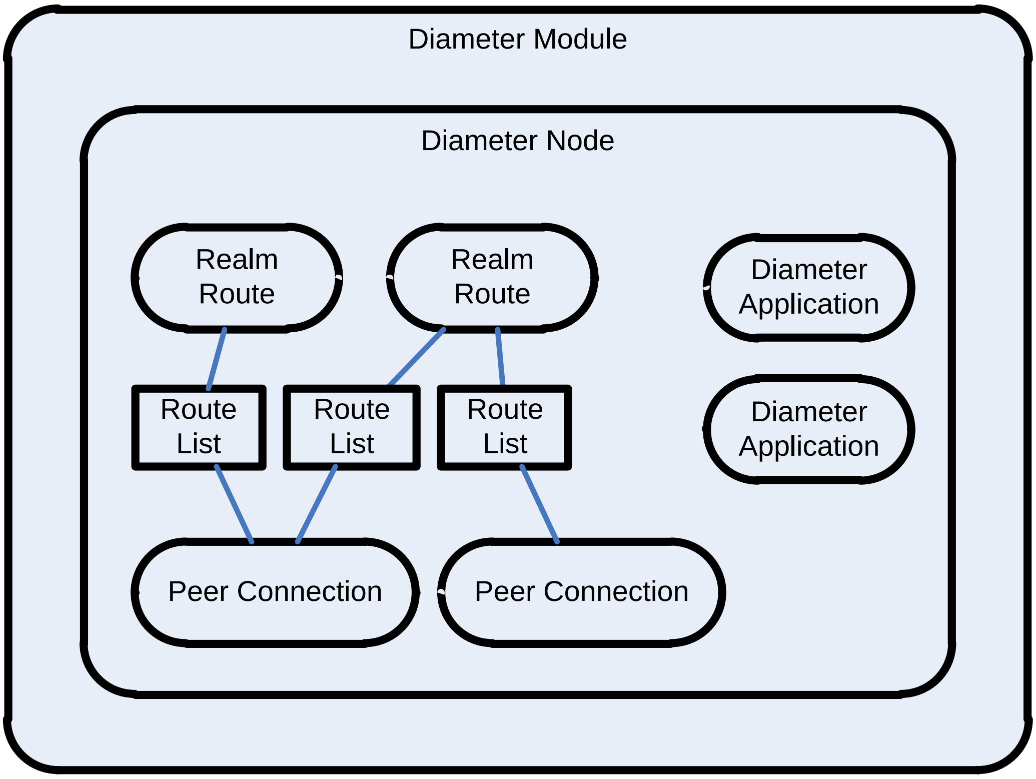

Route Selection
DMR_SESSION_REQ
Does request include Policy_ID?
Yes
No
Select Correct Normal Routing Table
Select Correct Policy ID Routing Table
No
Search for route based on Host, Realm, Application
Search for route based on Host, Realm, Application
Was Route Found?
Was Route Found?
Yes
Yes
Return No Route Error
Return Found Route
Return Found Route

Figure 4: Route Selection Flow

Note: Route Configuration and Selection operates independently for different Network Contexts.

3 Configuration Model

3.1 Module Configuration



The module configuration consists of the following principal elements:

Diameter Module: The DSI Diameter Module (DMR) receives configuration to control run time resource allocation and configure communication with supporting modules.

Diameter Node: The module may simultaneously support several Diameter Nodes each of which is independently visible in the Diameter network(s). Each node has its own User-Name, Origin-Realm, Origin-Host and Peer Connections.

Diameter Application: Each Diameter Node optionally supports a number of Diameter Applications for which it can receive traffic. These will be advertised during Capabilities Exchange.

Peer Connection: A Peer connection encompasses the SCTP transport connection to a remote Peer and has a Peer State Machine instance supporting Capability exchange, Watchdog, Transaction buffering etc.

Realm Route: A Realm Route defines a set of Diameter Messages for routing purposes. The set is identified based on parameters including Destination-Realm, Destination-Host and Application ID. A realm route may have a number of Route Lists.

Route List: This identifies a Peer Connection to which messages belonging to a Realm Route may be forwarded.

3.2 System Configuration

The DSI Diameter Module can be configured as part of a complete signaling system using the s7_mgt utility supplied as part of the Dialogic DSI Development Package. This utility can generate the required configuration messages from an appropriate configuration file. See *Dialogic® DSI Components – Software Environment Programmer's Manual* for further details on the s7_mgt utility and the configuration syntax used.

3.3 Configuration Sequence

The following is the recommend configuration sequence for the module.

1. Configure the main module settings.

   This will configure the module and define the module ids for the interfaces it will use to communicate with other parts of the system. It also allows module wide settings to be selected.

   (See DMR_MSG_CONFIG - Diameter Configuration Message. The message is generated by s7_mgt if the command SNSLI defines one or more Diameter associations).

2. Configure one or more Network Contexts.

   At least one of these is required and additional Network Contexts are required to connect to each distinct network. It defines settings that are specific to an individual network.

   (See DMR_MSG_NC_CFG - Diameter NC Configuration. The equivalent s7_mgt command is DMNCI as defined in the Software Environment Programmer's Manual).

3. Configure each local application

   Defines the local application to be supported for the purposes of capability negotiation. Also defines the mapping between the well-known AVP values for applications and a local application identifier.

   (See DMR_MSG_APP_CFG - Diameter Application Configuration. The equivalent s7_mgt command is DMAPI as defined in the Software Environment Programmer's Manual).

4. Configure one or more Peer Connections

   Defines a remote peer connection entity within a previously configured Network Context.

   (See DMR_MSG_PEER_CFG - Diameter Peer Configuration. The equivalent s7_mgt command is DMPRI as defined in the Software Environment Programmer's Manual).

5. Configure one or more Routes

   Defines the destination host and destination realm and links them to a Realm route.

(See DMR_MSG_ROUTE_CFG - Diameter Route Configuration. The equivalent s7_mgt command is DMRTI as defined in Software Environment Programmer’s Manual).

6. Configure one or more Route Lists

   Uses previously configured Realm Routes and Peer Connections.

   (See DMR_MSG_ROUTE_LIST_CFG - Diameter Route List Configuration. The equivalent s7_mgt command is DMRLI as defined in the Software Environment Programmer’s Manual).

4 Message Reference

4.1 Message Type Summary

The following table lists, by message type, the messages described in this manual.

Message Type	Mnemonic	Description
0x0465	DMR_MSG_EVENT_IND	Diameter Event Indication
0x0466	DMR_MSG_ERROR_IND	Diameter Error Indication
0x0468	DMR_MSG_PEER_STATE_IND	Peer Connection State Indication
0x6467	DMR_MSG_S_TRACE_MASK	Set Diameter Trace Masks
0x6470	DMR_MSG_R_PEER_STATE	Read Peer State Request
0x646e	DMR_MSG_R_NC_STATS	Diameter NC Statistics
0x6473	DMR_MSG_R_PEER_STATS	Diameter Peer Statistics
0x6474	DMR_MSG_R_APP_STATS	Diameter Application Statistics
0x6475	DMR_MSG_R_ROUTE_STATS	Diameter Route Statistics
0x6476	DMR_MSG_R_RL_STATS	Diameter Route List Statistics
0x7460	DMR_MSG_CONFIG	Diameter Configuration Message
0x7461	DMR_MSG_NC_CFG	Diameter NC Configuration
0x7462	DMR_MSG_APP_CFG	Application Configuration
0x7463	DMR_MSG_PEER_CFG	Diameter Peer Configuration
0x7464	DMR_MSG_ROUTE_CFG	Diameter Route Configuration
0x746f	DMR_MSG_ROUTE_LIST_CFG	Diameter Route List Configuration
0x7469	DMR_MSG_PEER_ACTIVATE	Diameter Peer Activate Request
0x746a	DMR_MSG_PEER_DEACTIVATE	Diameter Peer Deactivate Request
0x846d	DMR_MSG_SESSION_IND	Diameter Session Indication
0xc46c	DMR_MSG_SESSION_REQ	Diameter Session Request
0x7477	DMR_MSG_APP_REM	Diameter Application Removal
0x7478	DMR_MSG_PEER_REM	Diameter Peer Removal
0x7479	DMR_MSG_ROUTE_REM	Diameter Route Removal
0x747a	DMR_MSG_ROUTE_LIST_REM	Diameter Route List Removal
0x747b	DMR_MSG_NC_REM	Diameter Network Context Removal

4.2 User Interface Messages

Payload messages between DMR and the User Application are all exchanges in one of two message types – a Session Request which is generated by the Application and sent to DMR and a Session Indication which is sent from DMR to the Application.

Typically the user does not deal directly with these messages at the raw message level but makes use of the Diameter Functional API to create a correctly formatted message or recover the parameters from a message. Details of the Functional API for the Dialogic® DSI Diameter Stack is given in the *Diameter Functional API User Manual*.

This section describes the raw message format used on the interface between DMR and the User Application as follows:

- DMR_MSG_SESSION_REQ – Diameter Session Request
- DMR_MSG_SESSION_IND – Diameter Session Indication

4.2.1 Session Control Primitives - Parameter Area

The parameter area of the DMR_MSG_SESSION_REQ and DMR_MSG_SESSION_IND messages both use a Name-Length-Data structure within the parameter area to allow for variable size parameters and extensibility.

The Name-Length-Data method is used by several other Dialogic® DSI Stack Layers but the scheme used for DMR is different in that it always uses two bytes for the Parameter Name, two bytes for the Parameter Length and two bytes for the terminator zero.

The parameter area for each message is constructed as a single byte Primitive Name followed by a list of parameters (each in Name-Length-Data format), followed by a two byte terminator Name (set to zero) indicating that there are no further parameters.

The following diagrams illustrate the parameter area coding:

Primitive Type	Parameter	Parameter		Parameter	Zero

Each parameter is formatted as follows:

Name	Length	Data
2 byte	2 byte	'Parameter length' bytes

4.2.2 Session Control Primitives - Primitive Type

The coding of the Primitive Type octet uses in the Session Control Primitives is shown in the following table:

Primitive	Mnemonic	Value
DMR-OPEN	DMRSR_OPEN	1
DMR-CONTINUE	DMRSR_CONTIINUE	2
DMR-CLOSE	DMRSR_CLOSE	3
DMR-P-ABORT	DMRSR_P_ABORT	4
DMR-NOTIFY	DMRSR_NOTIFY	5

A typical User Initiated session will begin with Session Request containing an Open Primitive, which includes the Diameter Command. In the simplest case this will be ended by the reception of a Session Indication containing a Close Primitive which includes Answer to the Diameter Command.

If the Diameter module detects a failure case not initiated by the Peer then a Session Indication containing P-Abort is generated.

4.2.3 Session Control Primitives - Parameters

The following parameter names are defined for use in session primitive messages:

Parameter	Mnemonic	Value (dec)
Network context	DMRPN_nc	1
Diameter message	DMRPN_dmtr_msg	2
Provider reason	DMRPN_prov_rsn	3
Routing policy key	DMRPN_routing_policy_key	4
Notify reason	DMRPN_notify_rsn	5
User Action	DMRPN_user_action	6
Storage message	DMRPN_storage_msg	7

The coding for each parameter type is given in the following tables:

Parameter name	DMRPN_nc
Parameter length	Fixed, set to 2
Parameter data	Network Context Identifier. Defaults to zero if not supplied.

Parameter name	DMRPN_dmtr_msg
Parameter length	Variable. There is no specific max size of this parameter but the complete message containing this parameter must fit within the 4200 octet message structure.
Parameter data	A well-formed Diameter Message encoded compliant to RFC3588 starting with the Diameter Command Header and including all AVPs.

Parameter name	DMRPN_prov_rsn
Parameter length	Fixed, set to 2

Parameter data	Provider abort reason. Set to one of the following values.
	DMRPA_REQUEST_TIME_OUT = 1,
	DMRPA_GUARD_TIMER_TIMEOUT = 2,
	DMRPA_UNKNOWN_SSN = 3,
	DMRPA_FORMAT_ERR = 4,
	DMRPA_INVALID_REQ = 5,
	DMRPA_INVALID_COMMAND = 6,
	DMRPA_ROUTE_UNKNOWN = 7,
	DMRPA_ROUTE_UNAVAILABLE = 8,
	DMRPA_RESOURCE_UNAVAILABLE = 9,
	DMRPA_FAILURE_TO_SEND = 10

Parameter name	DMRPN_routing_policy_key
Parameter length	Fixed, set to 2
Parameter data	The policy identifier to be used to route the Diameter message. See section 2.9 Policy Based Routing.

Parameter name	DMRPN_notify_rsn
Parameter length	Fixed, set to 2

Parameter data	Reason for notice generation.
	DMRNR_FORMAT_ERR = 1,
	DMRNR_BAD_NC = 2,
	DMRNR_INVALID_REQ = 3,
	DMRNR_MISSING_PARAM = 4,
	DMRNR_COMMAND_PRS_ERR = 5,
	DMRNR_MISSING_AVP = 6,
	DMRNR_PENDING_RSP = 7,
	DMRNR_REQUEST_TIME_OUT = 8,
	DMRNR_FAILURE_TO_SEND = 9

Parameter name	DMRPN_user_action
Parameter length	Fixed, set to 4

Parameter data	User Action bit field.
	DMRUA_FLAGS_GRANT = 0x00000001
	DMRUA_FLAGS_DENY = 0x00000002
	DMRUA_FLAGS_STORE = 0x00000004
	DMRUA_FLAGS_SVC_ERROR = 0x00000008
	DMRUA_FLAGS_UNKNOWN = 0x00000010
	DMRUA_FLAGS_DELETE = 0x00000020

Parameter name	DMRPN_storage_msg
Parameter length	Variable. There is no specific max size of this parameter but the complete message containing this parameter must fit within the 4200 octet message structure.
Parameter data	For Session Requests this parameter is used in place of the DMRPN_dmtr_msg parameter to indicate a message that had previously been stored and should now be resent, For Session Indications this parameter may be present to indicate a message that should be stored by the user application for later resending. This parameter takes the same format as the DMRPN_dmtr_msg parameter and therefore is also a well-formed Diameter Message encoded compliant to RFC3588 starting with the Diameter Command Header and including all AVPs.

4.2.4 DMR_MSG_SESSION_REQ – Diameter Session Request

Synopsis

Message sent to the DMR module containing a Session Request Primitive.

Format

Message Header		
Field Name		**Meaning**
type		DMR_MSG_SESSION_REQ (0xc46c)
id		session ID
src		user application module id
dst		DMR_TASK_ID
rsp_req		0
hclass		0
status		0
err_info		0
len		Number of bytes of user data
Parameter Area		
Offset	**Size**	**Name**
0	1	Session primitive type octet.
1	Len - 3	Parameters in Name-Length-Data format.
Len-2	2	Set to zero to indicate end of message.

Description

This message is used by the User module to send session primitives to DMR.

All session primitives contain a Session ID, which is encoded in the message header. The Session ID is established by the module that generates the Open Primitive and used in all subsequent Session Primitives.

For an outgoing Session, the Diameter User is responsible for generating a Session ID. For an incoming Session DMR assigns the Session ID.

The following request primitive types are supported:

- Open

  Used to open a session.The request will also typically include a Diameter Command Request specified in the Diameter Message parameter (DMRPN_dmtr_msg). If the command was previously stored and now needs to be resent, then then command should be specified in the Storage Massage parameter (DMRPN_storage_msg).
- Continue

Used to request a session is maintained and also send either a Diameter Command Request or Answer.

- Close

  Used to request a session is closed. Typically also includes a Diameter Command Answer.

Parameter area contents:

The following table lists the parameters associates with each session request primitive and shows whether the parameter is Mandatory (M), in which case the message will be discarded is the parameter is omitted, or Optional (O), in which case the parameter is not considered essential.

Parameter	Primitive			
	Open (1) (Diameter Message)	**Open (1)** (Storage Message)	**Continue (2)**	**Close (3)**
Network Context	O	O	-	-
Diameter Message	M	-	M	O
Routing Policy Key	O	O	-	-
Storage Message	-	M	O	-

4.2.5 DMR_MSG_SESSION_IND – Diameter Session Indication

Synopsis

Message sent issued by DMR to the DMR-User containing a session indication primitive.

Format

Message Header		
Field Name		**Meaning**
type		DMR_MSG_SESSION_IND (0x846d)
id		session id
src		DMR_TASK_ID
dst		User application module id
rsp_req		0
hclass		0
status		0
err_info		0
len		Number of bytes of user data
Parameter Area		
Offset	**Size**	**Name**
0	1	Session primitive type octet.
1	Len -3	Parameters in Name-Length-Data format.
Len -2	2	Set to zero indicating end of message.

Description

This message is used by the DMR to send session primitives to the DMR-User.

All session primitives contain the Session ID of the session to which they belong. It is encoded in the message header.

The following indication primitive types are supported:

- Open

  Used to indicate a session has been opened. Typically the indication also includes a Diameter Command Request.

- Continue

  Used to indicate a session is still open and also includes either a Diameter Command Request or Answer.

- Close

  Used to request a session is closed. Typically also includes a Diameter Command Answer.

- P_Abort

Used to indicate a session has been aborted for reason given in the Provider Reason parameter.

- Notify

  Used to notify the application of an issue with a previous Session Request indicated by the Notify Reason parameter.

Parameters

The following table lists the parameters associates with each session indication primitive and shows whether the parameter is Mandatory (M), in which case the message will be discarded is the parameter is omitted, or Optional (O), in which case the parameter is not considered essential.

Parameter	Primitive				
	Open (1)	Continue (2)	Close (3)	P Abort (4)	Notify (5)
Network Context	M		-	-	-
Diameter Message	M	O	O	-	-
Provider Reason	O	-	O	M	-
Notify Reason	-	-	-	-	M
User Action	-	O	O	O	O
Storage Message	-	O	O	O	O

4.3 Configuration Messages

The Diameter Module (DMR) configuration is applied at system startup through the use of the following configuration messages. Typically these messages are generated at startup time by the configuration utility s7_mgt which takes the configuration from the config.txt file. The commands used in config.txt are documented in the *Software Environment Programmer's Manual.*

When s7_mgt is used for configuration, the message definitions in this section are not needed by the user.

The configuration must start with the DMR_MSG_CONFIG module configuration message and then one or more DMR_MSG_NC_CFG messages to configure the Diameter Node(s). Each node will then have one or more Diameter Applications registered (DMR_MSG_APP_CFG) in addition to Diameter peers and Realm Routes.

The following configuration messages are documented in this section:

- DMR_MSG_CONFIG - Diameter Configuration Message
- DMR_MSG_NC_CFG – Diameter NC Configuration
- DMR_MSG_APP_CFG – Diameter Application Configuration
- DMR_MSG_PEER_CFG – Diameter Peer Configuration
- DMR_MSG_ROUTE_CFG – Diameter Route Configuration
- DMR_MSG_ROUTE_LIST_CFG – Diameter Route List Configuration
- DMR_MSG_APP_REM – Diameter Application Removal
- DMR_MSG_PEER_REM – Diameter Peer Removal
- DMR_MSG_ROUTE_REM – Diameter Route Removal
- DMR_MSG_ROUTE_LIST_REM – Diameter Route List Removal
- DMR_MSG_NC_REM – Diameter NC Removal

4.3.1 Configuration Parameter Format

All configuration messages for DMR use a Name-Length-Data structure within the parameter area to allow for variable size parameters and extensibility.

The Name-Length-Data method is used by several other Dialogic® DSI Stack Layers but the scheme used for DMR is different in that it always uses two bytes for the Parameter Name, two bytes for the Parameter Length and two bytes for the terminator zero.

The parameter area for each configuration message is constructed as a list of parameters (each in Name-Length-Data format), followed by a two byte terminator Name (set to zero) indicating that there are no further parameters.

For each message the first configuration parameter contains the main fixed length configuration parameters for the message.

The following diagrams illustrate the parameter area coding:

Main Configuration Parameter	Additional Parameter	Additional Parameter		Additional Parameter	Zero

Each parameter is formatted as follows:

Name	Length	Data
2 byte	2 byte	‘Parameter length’ bytes

4.3.2 Configuration Parameter Definitions

Parameter	Mnemonic	Value (decimal)	Value (hex)	Type
Module Configuration	DMRCN_MOD_CONFIG	1	0x01	Compound parameter See section 4.3.3 for details
Network context	DMRCN_NC_CFG	2	0x02	Compound parameter See section 4.3.4 for details
Peer Configuration	DMRCN_PEER_CFG	3	0x03	Compound parameter See section 4.3.6 for details
Application Configuration	DMRCN_APP_CFG	4	0x04	Compound parameter See section 4.3.7 for details
Route Configuration	DMRCN_ROUTE_CFG	5	0x05	Compound parameter See section 4.3.7 for details
Route List Configuration	DMRCN_ROUTE_LIST_CFG	6	0x06	Compound parameter See section 4.3.8 for details
Origination Host	DMRCN_ORIGIN_HOST	7	0x07	String
Origination Realm	DMRCN_ORIGIN_REALM	8	0x08	String
Node Name	DMRCN_NODE_NAME	9	0x09	String
Peer Host	DMRCN_PEER_HOST	10	0x0a	String
Peer Realm	DMRCN_PEER_REALM	11	0x0b	String
Destination Host	DMRCN_DEST_HOST	12	0x0c	String
Destination Realm	DMRCN_DEST_REALM	13	0x0d	String
Event Information	DMRCN_EVENT_INFO	14	0x0e	Compound parameter See section 4.5.1 for details
Error Information	DMRCN_ERROR_INFO	15	0x0f	Compound parameter See section 4.5.2 for details
Array of app ids	DMRCN_APP_IDS	16	0x10	Compound parameter See section 4.5.3 for details
IP Address	DMRCN_IP_ADDR	17	0x11	See section 4.3.6 for details
Policy Id	DMRCN_POLICY_ID	18	0x12	See section 4.3.7 for details
Diameter Application Id	DMRCN_DMTR_APP_ID	19	0x13	See section 4.3.7 for details

4.3.3 DMR_MSG_CONFIG - Diameter Configuration Message

Synopsis

Message used to configure the Diameter module (DMR) for operation.

Format

Message Header		
Field Name		**Meaning**
type		DMR_MSG_CONFIG (0x7460)
id		0
src		Management Module Id
dst		DMR_TASK_ID
rsp_req		0
hclass		0
status		0
err_info		0
len		variable
Parameter Area		
Offset	**Size**	**Name**
0	Len -2	Parameters in Name-Length-Data format.
Len -2	2	Set to zero indicating end of message.

Description

This message is used to configure DMR for operation. It should be the first message sent to DMR and any messages received before a valid configuration message will be discarded. It should only be issued once.

The message parameters relate to the environment in which DMR is operating and the various capabilities which will be required of the module.

Parameters

DMR_MSG_CONFIG (0x7460)			
Parameter	**Mandatory/ Optional**	**Default Value**	**Notes**
DMRCN_MOD_CONFIG	M	n/a	

MOD_CONFIG

DMRCN_MOD_CONFIG parameter		
Offset	**Size**	**Name**
0	1	maint_id
1	1	trace_id
2	1	sctp_id
3	1	Reserved (set to zero)
4	2	max_peers
6	4	base_og_id
10	4	base_ic_id
14	4	nog_sessions
18	4	nic_sessions
22	4	max_throughput
26	4	options

maint_id

Maintenance module ID. This is the module to which event and error indications are sent.

trace_id

Trace module ID. This is the module to which any trace messages will be sent.

sctp_id

SCTP module ID. This is the module_id of the transport layer.

max_peers

The maximum number of Diameter peers that the module is required to support. This value is compared with any internal limits.

base_og_id

The first session ID for outgoing sessions that the user wishes to be handled by DMR. The subsequent (nog_sessions -1) session IDs will also be handled by the module. The user must ensure that the values used in the session ID field of all protocol messages pertaining to outgoing sessions lie within the correct range.

base_ic_id

The first session ID for incoming sessions that the user wishes to be handled by DMR. The subsequent (nic_sessions -1) session IDs will also be handled by the module. The Diameter Module allocates the session ID for each incoming session. It uses values in the range

nog_sessions

The maximum number of simultaneous outgoing sessions that the module is required to support. This value is compared with any internal limits.

nic_sessions

The maximum number of simultaneous incoming sessions that the module is required to support. This value is compared with any internal limits.

max_throughput

This parameter is reserved for future use and should be set to zero.

options

Run-time options field, reserved for future use set to zero

4.3.4 DMR_MSG_NC_CFG – Diameter NC Configuration

Synopsis

Message sent to DMR to configure a Diameter Network Context.

Format

<table>
<tr><th colspan="3">Message Header</th></tr>
<tr><th colspan="2">Field Name</th><th>Meaning</th></tr>
<tr><td colspan="2">Type</td><td>DMR_MSG_NC_CFG (0x7461)</td></tr>
<tr><td colspan="2">Id</td><td>nc_id</td></tr>
<tr><td colspan="2">Src</td><td>Sending module id</td></tr>
<tr><td colspan="2">Dst</td><td>DMR_TASK_ID</td></tr>
<tr><td colspan="2">rsp_req</td><td>0</td></tr>
<tr><td colspan="2">Hclass</td><td>0</td></tr>
<tr><td colspan="2">Status</td><td>0</td></tr>
<tr><td colspan="2">err_info</td><td>0</td></tr>
<tr><td colspan="2">Len</td><td>variable</td></tr>
<tr><th colspan="3">Parameter Area</th></tr>
<tr><th>Offset</th><th>Size</th><th>Name</th></tr>
<tr><td>0</td><td>Len -2</td><td>Parameters in Name-Length-Data format.</td></tr>
<tr><td>Len -2</td><td>2</td><td>Set to zero indicating end of message.</td></tr>
</table>

Description

The message allows the configuration of independent Diameter Network Contexts. Each Network Context embodies a separate Diameter Node each of which can be independently configured allowing different behaviour and routing rules for the module.

Parameters

<table>
<tr><td colspan="4">DMR_MSG_NC_CFG (0x7461)</td></tr>
<tr><th>Parameter</th><th>Mandatory/
Optional</th><th>Default Value</th><th>Notes</th></tr>
<tr><td>DMRCN_NC_CFG</td><td>M</td><td></td><td></td></tr>
<tr><td>DMRCN_ORIGIN_HOST</td><td>M</td><td></td><td></td></tr>
<tr><td>DMRCN_ORIGIN_REALM</td><td>M</td><td></td><td></td></tr>
<tr><td>DMRCN_NODE_NAME</td><td>M</td><td></td><td></td></tr>
</table>

NC_CFG

DMRCN_NC_CFG parameter		
Offset	**Size**	**Name**
0	4	options

nc_id

The logical id for this Network Context for use in communication with local management and maintenance agents.

node_name

The user_name value which should be used for Diameter protocol messages originated from this node. Formatted as a UTF-8 String.

origin_host

The origin_host value which should be used for Diameter protocol messages originated from this node. Formatted as a UTF-8 String.

origin_realm

The origin_realm value which should be used for Diameter protocol messages originated from this node. Formatted as a UTF-8 String.

options

Run-time options for this NC. None currently defined.

4.3.5 DMR_MSG_APP_CFG – Diameter Application Configuration

Synopsis

Message sent to DMR to configure a Diameter Application for a particular Network Context.

Format

Message Header		
Field Name		**Meaning**
type		DMR_MSG_APP_CFG (0x7462)
id		application_id
src		Sending module_id
dst		DMR_TASK_ID
rsp_req		0
hclass		0
status		0
err_info		0
len		variable
Parameter Area		
Offset	**Size**	**Name**
0	Len -2	Parameters in Name-Length-Data format.
Len -2	2	Set to zero indicating end of message.

Description

DMR allows Diameter Applications to be registered against a Network Context. Once configured Peer Connections part of the same Network Context will advertise this Application during Capabilities Exchange and messages received for the application will be forwarded to the configured User module_id.

Parameters

DMR_MSG_APP_CFG (0x7462)			
Parameter	**Mandatory/ Optional**	**Default Value**	**Notes**
DMRCN_APP_CFG	M		

APP_CFG

DMRCN_APP_CFG parameter		
Offset	**Size**	**Name**
0	4	options
4	1	mod_id
5	1	reserved (Must be set to 0)
6	2	mod_inst
8	2	nc_id
10	4	vendor_id
14	4	dmtr_app_id

application ID

The logical id for the application used in communication with DMR concerning this application. This value is of local significance between DMR and local configuration and maintenance agents. It is distinct and separate to the well-known 'app_id' value configured later in this message.

options

Run-time options for this application. Reserved for future use and should be set to zero.

mod_id

User module ID for this Application.

mod_inst

User Instance for this Application. This is of relevance to DMR-User applications deployed on remote RSI-Hosts.

nc_id

The Network Context Id this application is being registered with.

vendor_id

The Diameter Vendor ID AVP value for use in Diameter capability negotiation.

dmtr_app_id

The Diameter Application ID AVP value for use in Diameter capability negotiation.

4.3.6 DMR_MSG_PEER_CFG – Diameter Peer Configuration

Synopsis

Message sent to DMR configure a Diameter Peer connection for a particular Network Context.

Format

Message Header		
Field Name		**Meaning**
Type		DMR_MSG_PEER_CFG (0x7463)
Id		peer_id
Src		Sending module id
Dst		DMR_TASK_ID
rsp_req		0
Hclass		0
Status		0
err_info		0
Len		variable
Parameter Area		
Offset	**Size**	**Name**
0	Len -2	Parameters in Name-Length-Data format.
Len -2	2	Set to zero indicating end of message.

Description

The Diameter Module allows Peer Diameter nodes, with which it expects to communicate, to be statically configured. Peer nodes may be Diameter Agents, Servers or Clients and may use dynamically or statically configured transport connections. Each Peer Connection belongs to a pre-configured Network Context.

Parameters

DMR_MSG_PEER_CFG (0x7463)			
Parameter	**Mandatory/ Optional**	**Default Value**	**Notes**
DMRCN_PEER_CFG	M		
DMRCN_PEER_HOST	M		
DMRCN_PEER_REALM	M		
DMRCN_IP_ADDR	M		

PEER_CFG

DMRCN_PEER_CFG parameter		
Offset	**Size**	**Name**
0	4	options
4	2	nc_id
6	2	asc_id

peer_id

The logical id of this Peer used for control and communication of the Peer by local management and maintenance agents.

options

Run-time options for this Peer assigned according to the following table:

Bit	Mnemonic	Description
0	DMR_PEER_OPT_SERVER	Set if the transport connection end point acts as the Server side

nc_id

The logical id of a preconfigured Network Context to which this Peer belongs.

asc_id

The logical id of the transport connection used to communicate with this Peer. Messages sent to and received from the Transport Module (e.g. the SCTPN module) must be configured to use this logical id.

host

An ASCII or UTF-8 String identifying the expected host-name AVP value for this Peer. This can be used to correlate dynamically created transport connections with pre-configured Peers or to validate statically configured transport addresses advertise the expected host-name AVP value.

realm

An ASCII or UTF-8 String identifying the expected realm-name AVP value for this Peer. This can be used to correlate dynamically created transport connections with pre-configured Peers or to validate statically configured transport addresses advertise the expected realm-name AVP value.

ip_addr

The local IP addresses to be advertised to the Peer as part of the capabilities exchange. Allows up to 4 IP addresses, each 17 octets in length formatted as follows:

Octet	Field	Description
0	Format	Set to value 1 indicating an Ipv4 address.
1 .. 4	Network address	Octet 1 containing the most significant byte and octet 4 containing the least significant byte
5 .. 17	Reserved	Should be set to zero.

4.3.7 DMR_MSG_ROUTE_CFG – Diameter Route Configuration

Synopsis

Message sent to DMR to configure an entry in the Diameter Routing Table for a particular Network Context.

Format

Message Header		
Field Name		**Meaning**
Type		DMR_MSG_ROUTE_CFG (0x7464)
Id		route_id
Src		Sending module id
Dst		DMR_TASK_ID
rsp_req		0
Hclass		0
Status		0
err_info		0
Len		Variable
Parameter Area		
Offset	**Size**	**Name**
0	Len -2	Parameters in Name-Length-Data format.
Len -2	2	Set to zero indicating end of message.

Description

The Diameter Routing table allows the configuration of routing rules for use with Diameter Messages received from the User Application.

Selection of a Diameter Peer for message routing is based on network conditions and the following:

1) Peer route – Route selection is made based on a match of the Destination-Host protocol parameter
2) Realm route – Route selection is made based on a routing key match. The routing key consists of the Destination-Realm and Application-ID protocol parameters; the key also includes the user defined network context and routing policy key
3) Default route – User defined default route (per network context)

The message must contain either the DMRRT_OPT_DEFAULT run-time option or one of the optional parameters, DMRCN_DEST_HOST or DMRCN_DEST_REALM.

If the DMRCN_DEST_REALM parameter is present, DMRCN_DMTR_APP_ID and DMRCN_POLICY_ID may be used.

Parameters

DMR_MSG_ROUTE_CFG (0x7464)			
Parameter	**Mandatory/ Optional**	**Default Value**	**Notes**
DMRCN_ROUTE_CFG	M		
DMRCN_DEST_HOST	O		Peer based routing (Diameter Destination Host)
DMRCN_DEST_REALM	O		Realm based routing (Diameter Destination Realm)
DMRCN_DMTR_APP_ID	O		Realm based routing (Diameter Application-Id)
DMRCN_POLICY_ID	O		Realm based routing (User defined)

ROUTE_CFG

DMRCN_ROUTE_CFG parameter		
Offset	**Size**	**Name**
0	4	options
4	2	nc_id

options

Run-time options for this Peer assigned according to the following table:

Bit	Mnemonic	Description
0	DMRRT_OPT_DEFAULT	Designates this route as the Default Route for this Network Context. A default route will be used if an attempt to route does not match a host-based or realm-based route.
1	DMRRT_OPT_LOAD_SHARE	Load share traffic between available peer connections based on NAI value.

nc_id

The logical id of the Network Context for which this route applies.

DMRCN_DMTR_APP_ID

The Diameter Application Identifier for which this route applies.

DMRCN_DMTR_APP_ID parameter		
Offset	**Size**	**Name**
0	4	dmtr_app_id

DMRCN_POLICY_ID

A user defined identifier providing a routing policy for the route. The policy id is used to distinguish routes which otherwise share common routing parameters. The value can be used in to control route selection on a per-message basis.

DMRCN_POLICY_ID parameter		
Offset	**Size**	**Name**
0	2	dmtr_app_id

4.3.8 DMR_MSG_ROUTE_LIST_CFG – Diameter Route List Configuration

Synopsis

Message sent to DMR to configure a single route list entry. Each route list entry defines a mapping between a route and a peer for use in the Diameter Routing Table.

Format

Message Header		
Field Name		**Meaning**
Type		DMR_MSG_ROUTE_LIST_CFG (0x746f)
Id		route_list_id
Src		Sending module id
Dst		DMR_TASK_ID
rsp_req		0
Hclass		0
Status		0
err_info		0
Len		Variable
Parameter Area		
Offset	**Size**	**Name**
0	Len -2	Parameters in Name-Length-Data format.
Len -2	2	Set to zero indicating end of message.

Description

The Diameter Routing Table allows static configuration of Route Lists which associate a peer with a route. The peer defined in the Route List is a candidate next hop destination for the route.

A route may be associated with multiple peers and a peer may be associated with multiple routes.

Parameters

DMR_MSG_ROUTE_LIST_CFG (0x746f)			
Parameter	**Mandatory/ Optional**	**Default Value**	**Notes**
DMRCN_ROUTE_LIST_CFG	M		

ROUTE_CFG

DMRCN_ROUTE_LIST_CFG parameter		
Offset	**Size**	**Name**
0	4	options
4	2	route_id
6	2	peer_id

options

Run-time options. Reserved for future use and should be set to zero.

Route_id

The logical id of the route for which the peer_id will be associated

peer_id

The logical id of the peer which can be used as a next hop destination for the route_id.

4.3.9 DMR_MSG_APP_REM – Remove Diameter Application Configuration

Synopsis

Message sent to DMR to remove a Diameter Application for a particular Network Context.

Format

Message Header	
Field Name	**Meaning**
type	DMR_MSG_APP_REM (0x7477)
id	application_id
src	Sending module_id
dst	DMR_TASK_ID
rsp_req	0
hclass	0
status	0
err_info	0
len	0

Description

DMR allows Diameter Applications to be removed from a Network Context. Once removed, Peer Connections part of the same Network Context will no longer advertise this Application during Capabilities Exchange and messages received for the application will be no longer be forwarded to the user.

All peers within the network context must be deactivated prior to removing an application.

All routes within the network context which define a Diameter Application Id as part of its configuration must be removed before attempting to remove an application which shares the same Diameter Application Id.

4.3.10 DMR_MSG_PEER_REM – Remove Diameter Peer Configuration

Synopsis

Message sent to DMR to remove a Diameter Peer connection for a particular Network Context.

Format

Message Header	
Field Name	**Meaning**
type	DMR_MSG_PEER_REM (0x7478)
id	peer_id
src	Sending module id
dst	DMR_TASK_ID
rsp_req	0
hclass	0
status	0
err_info	0
len	0

Description

The Diameter Module allows Peer Diameter nodes, with which it expects to communicate, to be statically configured and removed. Peer nodes may be Diameter Agents, Servers or Clients and may use dynamically or statically configured transport connections. Each Peer Connection belongs to a pre-configured Network Context.

All route list entries which reference a peer must be removed prior to removing the peer configuration.

4.3.11 DMR_MSG_ROUTE_REM – Remove a Diameter Route Configuration

Synopsis

Message sent to DMR to remove an entry in the Diameter Routing Table for a particular Network Context.

Format

Message Header	
Field Name	**Meaning**
type	DMR_MSG_ROUTE_REM (0x7479)
id	route_id
src	Sending module id
dst	DMR_TASK_ID
rsp_req	0
hclass	0
status	0
err_info	0
len	0

Description

The Diameter Routing table may be updated by issuing this message to remove a routing entry. Once removed, user messages will no longer be matched to this route. The next available route matching the routing parameters of the user message will be selected.

All route list entries which reference a route must be removed prior to removing the route configuration.

4.3.12 DMR_MSG_ROUTE_LIST_REM – Remove a Diameter Route List Configuration

Synopsis

Message sent to DMR to remove a single route list entry. Each route list entry defines a mapping between a route and a peer for use in the Diameter Routing Table.

Format

Message Header	
Field Name	**Meaning**
type	DMR_MSG_ROUTE_LIST_REM (0x747a)
id	route_list_id
src	Sending module id
dst	DMR_TASK_ID
rsp_req	0
hclass	0
sStatus	0
err_info	0
len	0

Description

The Diameter Routing Table allows removal of Route List entries which associate a peer with a route. The peer defined in the Route List is a candidate next hop destination for the route.

4.3.13 DMR_MSG_NC_REM – Remove Diameter Network Context Configuration

Synopsis

Message sent to DMR to remove a Diameter Network Context configuration.

Format

Message Header	
Field Name	**Meaning**
type	DMR_MSG_APP_REM (0x747b)
id	nc_id
src	Sending module_id
dst	DMR_TASK_ID
rsp_req	0
hclass	0
status	0
err_info	0
len	0

Description

DMR allows Diameter Network Contexts to be removed. All resources currently configured within this network context such as Peers, Routes, Route Lists and Applications must be removed prior to removing the Network Context configuration.

4.4 Management Request Messages

This section details the format of the following management request messages that can be used to interface with DMR for control purposes or to request status.

- DMR_MSG_PEER_ACTIVATE – Diameter Peer Activate Request
- DMR_MSG_PEER_DEACTIVATE – Diameter Peer Deactivate Request
- DMR_MSG_S_TRACE_MASK – Set Diameter Trace Masks
- DMR_MSG_R_NC_STATS – Diameter NC Statistics
- DMR_MSG_R_PEER_STATE – Read Peer State Request
- DMR_MSG_R_PEER_STATS – Diameter Peer Statistics
- DMR_MSG_R_APP_STATS – Diameter Application Statistics
- DMR_MSG_R_ROUTE_STATS – Diameter Route Statistics
- DMR_MSG_R_RL_STATS – Diameter Route List Statistics

4.4.1 DMR_MSG_PEER_ACTIVATE – Diameter Peer Activate Request

Synopsis

This primitive is used by management to request DMR to activate the connection to the identified peer.

Format

Message Header	
Field Name	**Meaning**
type	DMR_MSG_PEER_ACTIVATE (0x7469)
id	peer_id
src	Sending module id
dst	DMR_TASK_ID
rsp_req	0
hclass	0
status	0
err_info	0
len	0

Description

On receipt of this message, DMR attempts to activate the specified Peer connection. Receipt of a confirmation message does not imply the connection is available for use merely that DMR is attempting to bring the connection into service. Peer availability is indicated through the DMR_MSG_PEER_STATE_IND message.

The user can determine the current state of a Peer connection on demand using the DMR_MSG_R_PEER_STATUS message.

4.4.2 DMR_MSG_PEER_DEACTIVATE – Diameter Peer Deactivate Request

Synopsis

This primitive is used by management to request DMR to deactivate the specified peer connection.

Format

Message Header	
Field Name	**Meaning**
type	DMR_MSG_PEER_DEACTIVATE (0x746a)
id	peer_id
src	Sending module id
dst	DMR_TASK_ID
rsp_req	0
hclass	0
status	0
err_info	0
len	0

Description

On receipt of this message DMR attempts to deactivate the specified Peer connection

4.4.3 DMR_MSG_S_TRACE_MASK – Set Diameter Trace Masks

Synopsis

Message used by management to set the DMR trace masks.

Format

<table>
<tr><th colspan="3">Message Header</th></tr>
<tr><th colspan="2">Field Name</th><th>Meaning</th></tr>
<tr><td colspan="2">type</td><td>DMR_MSG_S_TRACE_MASK (0x6467)</td></tr>
<tr><td colspan="2">id</td><td>0</td></tr>
<tr><td colspan="2">src</td><td>Sending module id</td></tr>
<tr><td colspan="2">dst</td><td>DMR_TASK_ID</td></tr>
<tr><td colspan="2">rsp_req</td><td>0</td></tr>
<tr><td colspan="2">hclass</td><td>0</td></tr>
<tr><td colspan="2">status</td><td>0</td></tr>
<tr><td colspan="2">err_info</td><td>0</td></tr>
<tr><td colspan="2">len</td><td>12</td></tr>
<tr><th colspan="3">Parameter Area</th></tr>
<tr><th>Offset</th><th>Size</th><th>Name</th></tr>
<tr><td>0</td><td>4</td><td>op_mask</td></tr>
<tr><td>4</td><td>4</td><td>ip_mask</td></tr>
<tr><td>8</td><td>4</td><td>mngt_mask</td></tr>
</table>

Description

The trace masks allow the user to request messages generated and received by the DMR module to be copied and traced to the trace module for diagnostic purposes. Tracing can be turned on and off dynamically on a per primitive basis. Tracing is controlled by three 32 bit masks (output, input and management) and bits set to one in the mask cause messages of that type to be traced.

Parameters

op_mask

The output event trace mask. This is a 32-bit value with bits set to 1 to cause a trace message to be sent to the trace module when DMR sends the associated protocol message.

Bit	Message to be Traced	Description
0	DMR_MSG_SESSION_IND	Trace session indication primitives issued to user by DMR.
1	DMR_MSG_EVENT_IND	Trace event indication issued by DMR.
2	DMR_MSG_ERROR_IND	Trace error indications issued by DMR.
3	SCTP_MSG_ACTIVATE	Trace SCTP activation requests issued by DMR.
4	SCTP_MSG_SHUTDOWN	Trace SCTP shutdown requests issued by DMR.
5	SCTP_MSG_TX_REQ	Trace SCTP transmit requests issued by DMR
7 to 31		Reserved for future use and should be set to zero.

Ip_mask

The input event trace mask. This is a 32-bit value with bits set to 1 to cause a trace message to be sent to the system trace module when DMR receives the associated protocol message.

Bit	Message to be Traced	Description
0	DMR_MSG_SESSION_REQ	Trace session request primitives received by DMR.
1	SCTP_MSG_RX_IND	Trace SCTP receive indications received by DMR.
2	SCTP_MSG_STATUS_CHANGE	Trace SCTP status messages received by DMR
3	SCTP_MSG_CONG_STATUS	Trace SCTP congestion indications received by DMR.
4 to 31		Reserved for future use and should be set to zero.

mngt_mask

The management event trace mask. This is a 32-bit value with bits set to 1 to cause a trace message to be sent to the system trace module when DMR receives the associated protocol message.

Bit	Message to be Traced	Description
0	DMR_MSG_S_TRACE_MASK	Trace any set trace mask primitives received by DMR.
1	DMR_MSG_NC_CFG	Trace NC configuration messages received by DMR.
2	DMR_MSG_PEER_CFG	Trace peer configuration messages received by DMR.
3	DMR_MSG_PEER_ACTIVATE	Trace peer activation requests received by DMR.
4	DMR_MSG_PEER_DEACTIVATE	Trace peer deactivation requests received by DMR.
5		Reserved for future use. Set to zero.
6	DMR_MSG_APP_CFG	Trace application configuration messages received by DMR.
7	DMR_MSG_ROUTE_CFG	Trace route configuration messages received by DMR.
8	DMR_MSG_R_NC_STATS	Trace read NC stats requests received by DMR.
9	DMR_MSG_R_PEER_STATE	Trace read peer state requests received by DMR.
10		Reserved for future use. Set to zero.
11	DMR_MSG_ROUTE_LIST_CFG	Trace route list configuration messages received by DMR.
12 to 31		Reserved for future use and should be set to zero.

4.4.4 DMR_MSG_R_NC_STATS – Diameter NC Statistics

Synopsis

Message used to query measurements for a specific Diameter Network Context.

Format

Message Header		
Field Name		**Meaning**
type		DMR_MSG_R_NC_STATS (0x646e)
id		nc_id
src		Sending module id
dst		DMR_TASK_ID
rsp_req		Sending layer's bit must be set
hclass		0
status		0 – Leave stats unchanged 1 – Reset stats after reading
err_info		0
len		16
Parameter Area		
Offset	**Size**	**Name**
0	4	ssn_alloc_count
4	4	ssn_dealloc_count
8	4	tran_alloc_count
12	4	tran_dealloc_count

Description

This message allows the user to query the Network Context to determine its current usage of system resources. The counters can be reset or left unchanged depending on the setting of the status field. The user should send the message with all fields set to zero.

Note: In this context a transaction is made up of a Diameter Request and a corresponding Diameter Answer. A session is made up of one or more transactions between two peer entities.

Parameters

ssn_alloc_count

The number of session resources which have been allocated.

Ssn_dealloc_count

The number of session resources which have been de-allocated.

Tran_alloc_count

The number of transaction resources which have been allocated.

Tran_dealloc_count

The number of transaction resources which have been de-allocated.

4.4.5 DMR_MSG_R_PEER_STATE – Read Peer State Request

Synopsis

Message used to query the current Peer State.

Format

Message Header		
Field Name		**Meaning**
type		DMR_MSG_R_PEER_STATE (0x6470)
id		peer_id
src		Sending module id
dst		DMR_TASK_ID
rsp_req		Sending layer's bit must be set
hclass		0
status		0
err_info		0
len		3
Parameter Area		
Offset	**Size**	**Name**
0	1	availability
1	2	State_flags

Description

This message allows the user to query the Peer State. The user should allocate a message with sufficient space in the parameter area and DMR will fill in the parameter area in the response message.

Parameters

availability

Peer availability state. Possible values are shown in the table below:

Value	Mnemonic	Description
0	DMR_PEER_STATE_UNKNOWN	Entered Unknown State
1	DMR_PEER_STATE_CLOSED	Entered CLOSED State
2	DMR_PEER_STATE_OPEN	Entered OPEN State

state_flags

Additional state information related to the peer. Bit 0 is set to 1 if a request to activate the peer has been made. All other bits are reserved for future use and will be set to zero.

4.4.6 DMR_MSG_R_PEER_STATS – Diameter Peer Statistics

Synopsis

Message used to query measurements for a specific Diameter Peer.

Format

Message Header		
Field Name		**Meaning**
type		DMR_MSG_R_PEER_STATS (0x6473)
id		peer_id
src		Sending module id
dst		DMR_TASK_ID
rsp_req		Sending layer's bit must be set
hclass		0
status		0 – Leave stats unchanged 1 – Reset stats after reading
err_info		0
len		32
Parameter Area		
Offset	**Size**	**Name**
0	4	Period
4	4	tx_octet
8	4	rx_octet
12	4	tx_msg
16	4	rx_msg
20	4	peer_error
24	4	closed_cnt
28	4	open_duration

Description

This message allows the user to query a specific peer to read measurements. The counters can be reset or left unchanged depending on the setting of the status field. The user should send the message with all fields set to zero.

Parameters

period

Period over which the measurements have been collected (in multiples of 100ms).

Tx_octet

Number of octets transmitted.

Rx_octet

Number of octets received.

Tx_msg

Number of messages transmitted.

Rx_msg

Number of messages received.

Peer_error

Number of errors associated with this peer including peer timeouts, unexpected messages and remote disconnections.

Closed_cnt

Count of state transitions to the CLOSED state.

Open_duration

Duration within period the Peer has been in the OPEN state (in multiples of 100ms).

4.4.7 DMR_MSG_R_APP_STATS – Diameter Application Statistics

Synopsis

Message used to query measurements for a specific Diameter Application.

Format

<table>
<tr><th colspan="3">Message Header</th></tr>
<tr><th colspan="2">Field Name</th><th>Meaning</th></tr>
<tr><td colspan="2">type</td><td>DMR_MSG_R_APP_STATS (0x6474)</td></tr>
<tr><td colspan="2">id</td><td>application_id</td></tr>
<tr><td colspan="2">src</td><td>Sending module id</td></tr>
<tr><td colspan="2">dst</td><td>DMR_TASK_ID</td></tr>
<tr><td colspan="2">rsp_req</td><td>Sending layer’s bit must be set</td></tr>
<tr><td colspan="2">hclass</td><td>0</td></tr>
<tr><td colspan="2">status</td><td>0 – Leave stats unchanged
1 – Reset stats after reading</td></tr>
<tr><td colspan="2">err_info</td><td>0</td></tr>
<tr><td colspan="2">len</td><td>20</td></tr>
<tr><th colspan="3">Parameter Area</th></tr>
<tr><th>Offset</th><th>Size</th><th>Name</th></tr>
<tr><td>0</td><td>4</td><td>period</td></tr>
<tr><td>4</td><td>4</td><td>tx_octet</td></tr>
<tr><td>8</td><td>4</td><td>rx_octet</td></tr>
<tr><td>12</td><td>4</td><td>tx_msg</td></tr>
<tr><td>16</td><td>4</td><td>rx_msg</td></tr>
</table>

Description

This message allows the user to query a specific application to read measurements. The counters can be reset or left unchanged depending on the setting of the status field. The user should send the message with all fields set to zero.

Parameters

period

Period over which the measurements have been collected (in multiples of 100ms).

Tx_octet

Number of octets transmitted.

Rx_octet
Number of octets received.

Tx_msg
Number of messages transmitted.

Rx_msg
Number of messages received.

4.4.8 DMR_MSG_R_ROUTE_STATS – Diameter Route Statistics

Synopsis

Message used to query measurements for a specific Diameter Route.

Format

<table>
<tr><th colspan="3">Message Header</th></tr>
<tr><th colspan="2">Field Name</th><th>Meaning</th></tr>
<tr><td colspan="2">type</td><td>DMR_MSG_R_ROUTE_STATS (0x6475)</td></tr>
<tr><td colspan="2">id</td><td>route_id</td></tr>
<tr><td colspan="2">src</td><td>Sending module id</td></tr>
<tr><td colspan="2">dst</td><td>DMR_TASK_ID</td></tr>
<tr><td colspan="2">rsp_req</td><td>Sending layer’s bit must be set</td></tr>
<tr><td colspan="2">hclass</td><td>0</td></tr>
<tr><td colspan="2">status</td><td>0 – Leave stats unchanged
1 – Reset stats after reading</td></tr>
<tr><td colspan="2">err_info</td><td>0</td></tr>
<tr><td colspan="2">len</td><td>16</td></tr>
<tr><th colspan="3">Parameter Area</th></tr>
<tr><th>Offset</th><th>Size</th><th>Name</th></tr>
<tr><td>0</td><td>4</td><td>period</td></tr>
<tr><td>4</td><td>4</td><td>tx_octet</td></tr>
<tr><td>8</td><td>4</td><td>tx_msg</td></tr>
<tr><td>12</td><td>4</td><td>redirect</td></tr>
</table>

Description

This message allows the user to query a specific Diameter route to read measurements. The counters can be reset or left unchanged depending on the setting of the status field. The user should send the message with all fields set to zero.

Parameters

period

Period over which the measurements have been collected (in multiples of 100ms).

Tx_octet

Number of user provided octets transmitted over this route.

Tx_msg

Number of user provided messages transmitted over this route.

Redirect
Number of Diameter messages redirected to an alternative peer.

4.4.9 DMR_MSG_R_RL_STATS – Diameter Route List Statistics

Synopsis

Message used to query measurements for a specific Diameter Route List entry.

Format

<table>
<tr><th colspan="3">Message Header</th></tr>
<tr><th colspan="2">Field Name</th><th>Meaning</th></tr>
<tr><td colspan="2">type</td><td>DMR_MSG_R_RL_STATS (0x6476)</td></tr>
<tr><td colspan="2">id</td><td>route_list_id</td></tr>
<tr><td colspan="2">src</td><td>Sending module id</td></tr>
<tr><td colspan="2">dst</td><td>DMR_TASK_ID</td></tr>
<tr><td colspan="2">rsp_req</td><td>Sending layer's bit must be set</td></tr>
<tr><td colspan="2">hclass</td><td>0</td></tr>
<tr><td colspan="2">status</td><td>0 – Leave stats unchanged
1 – Reset stats after reading</td></tr>
<tr><td colspan="2">err_info</td><td>0</td></tr>
<tr><td colspan="2">len</td><td>12</td></tr>
<tr><th colspan="3">Parameter Area</th></tr>
<tr><th>Offset</th><th>Size</th><th>Name</th></tr>
<tr><td>0</td><td>4</td><td>period</td></tr>
<tr><td>4</td><td>4</td><td>down_count</td></tr>
<tr><td>8</td><td>4</td><td>up_duration</td></tr>
</table>

Description

This message allows the user to query a specific Diameter Route List entry to read measurements. The counters can be reset or left unchanged depending on the setting of the status field. The user should send the message with all fields set to zero.

Parameters

period

Period over which the measurements have been collected (in multiples of 100ms).

Down_count

Count of Route List state transitions to the DOWN state.

Up_duration

Duration within period the Route List has been in the UP state (in multiples of 100ms).

4.5 Event Indication Messages

Event indication messages are the mechanism by which protocol and software error events are reported to management.

The following event indication messages generated by DMR are documented in this section:

- DMR_MSG_EVENT_IND – Diameter Event Indication
- DMR_MSG_ERROR_IND – Diameter Error Indication
- DMR_MSG_PEER_STATE_IND - Peer Connection State Indication

4.5.1 DMR_MSG_EVENT_IND – Diameter Event Indication

Synopsis

Message sent by DMR to the management module to indicate a protocol related event has occurred.

Format

Message Header		
Field Name		**Meaning**
type		DMR_MSG_EVENT_IND (0x0465)
id		See below
src		DMR_TASK_ID
dst		Maintenance module id
rsp_req		0
hclass		0
status		Event Code (see below)
err_info		0
len		8
Parameter Area		
Offset	**Size**	**Name**
0	4	param1
4	4	param2

Description

Message used by DMR to indicate a Diameter protocol event has occurred.

Parameters

Event Code

The event code contained in the status field of the message indicates the type of event. Possible values are listed in the following table that also lists the meaning of the id field, param1 and param2 in each case. Unused values are set to zero.

Code	Mnemonic	Id	Param1	Param2	Description
1	DMR_EVENT_DECODE_ AVP_MISSING	0	int_status		A mandatory AVP is missing from a diameter message.
2	DMR_EVENT_DECODE_ AVP_DOESNT_BELONG	int_status	0	0	An inappropriate AVP has been detected whilst decoding

Code	Mnemonic	Id	Param1	Param2	Description
3	DMR_EVENT_DECODE_AVP_MISFORMATTED	nc_id	AVP id	len	An AVP has been decode with an invalid format, e.g. Bad length
4	DMR_EVENT_DECODE_AVP_ERROR	nc_id	AVP code	bytes_processed	A badly formatted Diameter Message has been received. Param2 indicates the offset of the error within the Diameter message
5	DMR_EVENT_DECODE_MSG_ERROR	session_id	0	0	Decoding of message detected a problem
6	DMR_EVENT_DECODE_PEER_NAME	peer_id	0	0	Could not decode Peer name as part of Capability Exchange.
7	DMR_EVENT_DECODE_PEER_REALM	peer_id	0	0	Could not decode Peer realm as part of Capability Exchange.
10	DMR_EVENT_NC_UNKNOWN	0	nc_id		The identified network context ID is invalid.
11	DMR_EVENT_NC_UNKNOWN_APP_ID	nc_id	app_id	0	Could not identify application based on an unknown application id.
12	DMR_EVENT_NC_APP_ID	0	0	0	
20	DMR_EVENT_PEER_ERROR	peer_id	num_app	0	An error adding an application to a peer has occurred.
21	DMR_EVENT_PEER_REFUSED	peer_id	0	0	A dynamic connection request has been refused
22	DMR_EVENT_PEER_UNEX_REALM	peer_id	0	0	The peers advertised realm name does not match that configured.
23	DMR_EVENT_PEER_UNEX_HOST	peer_id	0	0	The peers advertised host name does not match that configured.
24	DMR_EVENT_PEER_UNEX_ASC_ID	assoc_id	0	0	Unexpected association for peer

Code	Mnemonic	Id	Param1	Param2	Description
25	DMR_EVENT_PEER_CAP_EX_NO_COMMON_APPS	peer_id	0	0	A capability exchange procedure failed due to no common applications
26	DMR_EVENT_PEER_CAP_EX_NO_COMMON_SECURITY	peer_id	0	0	A capability exchange procedure failed due to no common security
27	DMR_ERROR_PEER_CAP_EX_FAILED	peer_id	0	0	A capability exchange procedure failed
28	DMR_EVENT_PEER_DPR_RXED	peer_id	Disconnection-cause AVP value	0	A DPR message has been received
29	DMR_EVENT_PEER_UNEX_APP_ID	peer_id	Application-Id	0	An unexpected Application-Id has been received.
30	DMR_EVENT_TRAN_RUN_OUT	0	0	0	The system failed to allocate a new transaction resource
40	DMR_EVENT_SSN_BAD_ID	session_id	0	0	An event has been received for a session id which is not in use
41	DMR_EVENT_SSN_INVALID_ID	session_id	0	0	An event has been received for a session id which is invalid
42	DMR_EVENT_SSN_RUN_OUT	0	0	0	The module has run out of session resources to allocate
51	DMR_EVENT_REQ_BAD_PRIM_TYPE	session_id	0	0	User request contained an unknown primitive type.
52	DMR_EVENT_REQ_MISSING_PARAM		param tag	primitive type	A mandatory parameter is missing from the request
53	DMR_EVENT_REQ_DMTR_REQ_FLAG	session_id	0	0	The Diameter request flag is set - reset expected
54	DMR_EVENT_REQ_BAD_DMTR_MSG	session_id	nc_id	0	A problem was found parsing with the request values

Code	Mnemonic	Id	Param1	Param2	Description
55	DMR_EVENT_REQ_BAD_APP_ID	0	nc_id	application_id	Session request received for an unknown and unhandled application
56	DMR_EVENT_REQ_TOO_SHORT	session_id	len	0	The DMR_SESSION_REQ message was too short
57	DMR_EVENT_REQ_BAD_PARAM	session_id	param_tag	len	A problem was found parsing the request format
58	DMR_EVENT_REQ_BAD_FMT	session_id	0	0	The parameter was an invalid length
59	DMR_EVENT_REQ_BAD_TERM	session_id	0	0	The terminating two bytes should have value zero.
60	DMR_EVENT_REQ_UNEX_PRIM	session_id	0	0	Unexpected primitive type received for the session in use.
61	DMR_EVENT_REQ_NO_DEST_AVP	session_id	0	0	Request has no host or realm specified
62	DMR_EVENT_REQ_NO_RESULT_AVP	session_id	0	0	Request has no result or experimental result code specified
63	DMR_EVENT_REQ_MISSING_AVP	session_id	0	0	A required AVP for message is missing
70	DMR_EVENT_ROUTE_UNKNOWN	session_id	nc_id	0	An appropriate route could not be selected
71	DMR_EVENT_ROUTE_UNAVAILABLE	rte_id	session_id	0	The Route selected is currently unavailable for routing.

4.5.2 DMR_MSG_ERROR_IND – Diameter Error Indication

Synopsis

Message sent by DMR to the management module to indicate an exceptional or erroneous protocol event or implementation error has occurred.

Format

<table>
<tr><th colspan="3">Message Header</th></tr>
<tr><th colspan="2">Field Name</th><th>Meaning</th></tr>
<tr><td colspan="2">type</td><td>DMR_MSG_ERROR_IND (0x0466)</td></tr>
<tr><td colspan="2">id</td><td>See below</td></tr>
<tr><td colspan="2">src</td><td>DMR_TASK_ID</td></tr>
<tr><td colspan="2">dst</td><td>Maintenance module id</td></tr>
<tr><td colspan="2">rsp_req</td><td>0</td></tr>
<tr><td colspan="2">hclass</td><td>0</td></tr>
<tr><td colspan="2">status</td><td>error_code (see below)</td></tr>
<tr><td colspan="2">err_info</td><td>0</td></tr>
<tr><td colspan="2">len</td><td>8</td></tr>
<tr><th colspan="3">Parameter Area</th></tr>
<tr><th>Offset</th><th>Size</th><th>Name</th></tr>
<tr><td>0</td><td>4</td><td>param1</td></tr>
<tr><td>4</td><td>4</td><td>param2</td></tr>
</table>

Description

This message used by DMR to convey the occurrence of an exceptional or erroneous protocol event or implementation error and to convey additional diagnostic information to facilitate investigation of the problem.

Parameters

error_code

The error code contained in the status filed of the message indicates the type of error. Possible values are listed in the following table that also lists the meaning of the id field, param1 and param2 in each case. Unused values are set to zero.

Code	Mnemonic	Id	param1	param2	Description
1	DMR_ERROR_PEER_CREATION_FAILED	peer_id	int_status	0	Creation of a Peer could not be performed
2	DMR_ERROR_PEER_UNKNWN_ERROR,	peer_id	0	0	An event occurred for an unknown peer
3	DMR_ERROR_PEER_NOT_IN_USE	peer_id	0	0	An event unexpectedly occurred for a peer not in use
10	DMR_ERROR_SSN_OOB_EVENT	session_id	0	0	An event has occurred for a session id that is not in use or out of bounds
11	DMR_ERROR_SSN_CREATION_FAILED	session_id	int_status	0	A session could not be created
12	DMR_ERROR_SSN_ANS_TX_FAILED	session_id	Int_status	0	An attempt to send Answer message has failed
13	DMR_ERROR_SSN_DESTRUCT_FAILED	session_id	int_status	0	An attempt to close a session failed
14	DMR_ERROR_SSN_UNEX_EVENT	session_id	session_type	0	An event has been received which is unexpected for the session type
20	DMR_ERROR_ROUTE_BAD_PEER_ID	route_id	peer_id	0	Request processing failed due to an inappropriate peer_id
30	DMR_ERROR_UNEXP_EVENT	nc_id	0	app_id	An unexpected event was detected during session processing

4.5.3 DMR_MSG_PEER_STATE_IND - Peer Connection State Indication

Synopsis

Message issued by DMR to advise management of changes in state of the Peer connection.

Format

Message Header		
Field Name		**Meaning**
type		DMR_MSG_PEER_STATE_IND (0x0468)
id		peer_id
src		DMR_TASK_ID
dst		Maintenance module id
rsp_req		0
hclass		0
status		peer_state (see below)
err_info		0
len		variable
Parameter Area		
Offset	**Size**	**Name**
0	Len -2	Parameters in Name-Length-Data format.
Len -2	2	Set to zero indicating end of message.

Description

This primitive is used by DMR to advise management of changes of state of the Peer Connection. This message is intended for diagnostic and maintenance purposes and does not form part of the protocol specified primitives.

Parameters

peer_state

The new state of the Peer connection. The following table shows the possible values and their meanings:

Value	Mnemonic	Description
0	DMR_PEER_STATE_UNKNOWN	Entered Unknown State
1	DMR_PEER_STATE_CLOSED	Entered CLOSED State
2	DMR_PEER_STATE_OPEN	Entered OPEN State

APP_IDS

An array of App Ids which are appropriate for this Indication.

DMRCN_APP_IDS parameter		
Offset	**Size**	**Name**
0	2	App id (First in array)
…		
(N - 1) * 2	2	App Id (Nth in array)

5 External Message Definitions

The Dialogic® DSI Diameter Stack is intended for use in conjunction with the Dialogic® DSI SIGTRAN Stack SCTP Layer. Specifically, it is intended for use with the SCTPN binary which makes use of the Native SCTP stack within the host operating system. It provides a DSI message based interface which is used by the DSI Diameter Stack for the purpose of controlling the SCTP associations and transferring date to the peer node.

Detail of the interface to the SCTPN binary is contained within the *Dialogic® DSI SCTP Programmer's Manual*.

The following primitives are used by the DSI Diameter Stack:

Primitive	**Message Type**	**Description**
SCTP Activate association	0x728a	Used by Diameter to activate an SCTP association.
SCTP Shutdown association	0x728b	Used by Diameter to gracefully close down an SCTP association
SCTP Abort association	0x728c	Used by Diameter to initiate the abort of an SCTP association
SCTP Data transfer request	0xc280	Used by Diameter to transmit a data packet over an association.
SCTP Status change	0x028d	Indication from SCTP of a change in association status.
SCTP Data transfer indication	0x8281	Indication from SCTP of a received data packet.
SCTP Congestion status	0x028f	Indication from SCTP of a change in the congestion status of the association
SCTP Network status	0x028e	Indication from SCTP of a change in status a network address within an association.