Dialogic®
Making Innovation Thrive™

Dialogic® NaturalAccess™ Fusion NbUP Endpoint and Clear Channel Reference Manual

October 2010 64-1133-01 Rev A

www.dialogic.com

Copyright and legal notices

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 Corporation or its 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 medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

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 Corporation 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 9800 Cavendish Blvd., 5th Floor, Montreal, Quebec, Canada H4M 2V9. 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.

Any use case(s) shown and/or described herein represent one or more examples of the various ways, scenarios or environments in which Dialogic® products can be used. Such use case(s) are non-limiting and do not represent recommendations of Dialogic as to whether or how to use Dialogic products.

Dialogic, Dialogic Pro, Brooktrout, Diva, Cantata, SnowShore, Eicon, Eicon Networks, NMS Communications, NMS (stylized), Eiconcard, SIPcontrol, Diva ISDN, TruFax, Exnet, EXS, SwitchKit, N20, Making Innovation Thrive, Connecting to Growth, Video is the New Voice, Fusion, Vision, PacketMedia, NaturalAccess, NaturalCallControl, NaturalConference, NaturalFax and Shiva, among others as well as related logos, are either registered trademarks or trademarks of Dialogic Corporation or its 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 9800 Cavendish Blvd., 5th Floor, Montreal, Quebec, Canada H4M 2V9. 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.

Revision History

Revision	Release date	Notes
9000-62722-01	September 2008	PLR, See original document for dates and revision notes.
64-1133-01 Rev A	October 2010	LBG, Video Access 3.3
Last modified: 2010-10-15		

Refer to www.dialogic.com for product updates and for information about support policies, warranty information, and service offerings.

Table of Contents

1. Introduction

The *Dialogic® NaturalAccess™ Fusion NbUP Endpoint and Clear Channel Reference Manual* provides detailed information on newly introduced MSPP components namely NbUP endpoint and Clear channel that are needed to terminate 3GPP Rel4 IP User Plane.

This manual describes the features of NbUP endpoint and Clear channel along with the other MSPP components that were updated to support NbUP protocol.

Terminology

Note: The product to which this document pertains is part of the NMS Communications Platforms business that was sold by NMS Communications Corporation ("NMS") to Dialogic Corporation ("Dialogic") on December 8, 2008. Accordingly, certain terminology relating to the product has been changed. Below is a table indicating both terminology that was formerly associated with the product, as well as the new terminology by which the product is now known. This document is being published during a transition period; therefore, it may be that some of the former terminology will appear within the document, in which case the former terminology should be equated to the new terminology, and vice versa.

Former terminology	Dialogic terminology
CG 6060 Board	Dialogic® CG 6060 PCI Media Board
CG 6060C Board	Dialogic® CG 6060C CompactPCI Media Board
CG 6565 Board	Dialogic® CG 6565 PCI Media Board
CG 6565C Board	Dialogic® CG 6565C CompactPCI Media Board
CG 6565e Board	Dialogic® CG 6565E PCI Express Media Board
CX 2000 Board	Dialogic® CX 2000 PCI Station Interface Board
CX 2000C Board	Dialogic® CX 2000C CompactPCI Station Interface Board
AG 2000 Board	Dialogic® AG 2000 PCI Media Board
AG 2000C Board	Dialogic® AG 2000C CompactPCI Media Board
AG 2000-BRI Board	Dialogic® AG 2000-BRI Media Board
NMS OAM Service	Dialogic® NaturalAccess™ OAM API
NMS OAM System	Dialogic® NaturalAccess™ OAM System
NMS SNMP	Dialogic® NaturalAccess™ SNMP API

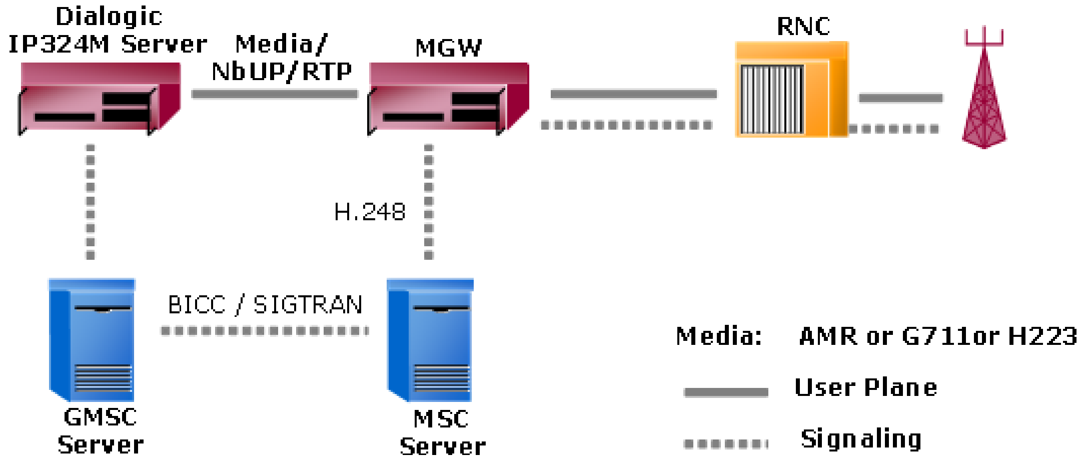
Former terminology	Dialogic terminology
Natural Access	Dialogic® NaturalAccess™ Software
Natural Access Service	Dialogic® NaturalAccess™ Service
Fusion	Dialogic® NaturalAccess™ Fusion™ VoIP API
ADI Service	Dialogic® NaturalAccess™ Alliance Device Interface API
CDI Service	Dialogic® NaturalAccess™ CX Device Interface API
Digital Trunk Monitor Service	Dialogic® NaturalAccess™ Digital Trunk Monitoring API
MSPP Service	Dialogic® NaturalAccess™ Media Stream Protocol Processing API
Natural Call Control Service	Dialogic® NaturalAccess™ NaturalCallControl™ API
NMS GR303 and V5 Libraries	Dialogic® NaturalAccess™ GR303 and V5 Libraries
Point-to-Point Switching Service	Dialogic® NaturalAccess™ Point-to-Point Switching API
Switching Service	Dialogic® NaturalAccess™ Switching Interface API
Voice Message Service	Dialogic® NaturalAccess™ Voice Control Element API
NMS CAS for Natural Call Control	Dialogic® NaturalAccess™ CAS API
NMS ISDN	Dialogic® NaturalAccess™ ISDN API
NMS ISDN for Natural Call Control	Dialogic® NaturalAccess™ ISDN API
NMS ISDN Messaging API	Dialogic® NaturalAccess™ ISDN Messaging API
NMS ISDN Supplementary Services	Dialogic® NaturalAccess™ ISDN API Supplementary Services
NMS ISDN Management API	Dialogic® NaturalAccess™ ISDN Management API
NaturalConference Service	Dialogic® NaturalAccess™ NaturalConference™ API
NaturalFax	Dialogic® NaturalAccess™ NaturalFax™ API

Former terminology	Dialogic terminology
SAI Service	Dialogic® NaturalAccess™ Universal Speech Access API
NMS SIP for Natural Call Control	Dialogic® NaturalAccess™ SIP API
NMS RJ-45 interface	Dialogic® MD1 RJ-45 interface
NMS RJ-21 interface	Dialogic® MD1 RJ-21 interface
NMS Mini RJ-21 interface	Dialogic® MD1 Mini RJ-21 interface
NMS Mini RJ-21 to NMS RJ-21 cable	Dialogic® MD1 Mini RJ-21 to MD1 RJ-21 cable
NMS RJ-45 to two 75 ohm BNC splitter cable	Dialogic® MD1 RJ-45 to two 75 ohm BNC splitter cable
NMS signal entry panel	Dialogic® Signal Entry Panel
Video Access Utilities	Dialogic® NaturalAccess™ Video Access Toolkit Utilities
Video Mail Application Demonstration Program	Dialogic® NaturalAccess™ Video Access Toolkit Video Mail Application Demonstration Program
Video Messaging Server Interface	Dialogic® NaturalAccess™ Video Access Toolkit Video Messaging Server Interface
3G-324M Interface	Dialogic® NaturalAccess™ Video Access Toolkit 3G-324M Interface

2. Fusion NbUP Endpoint

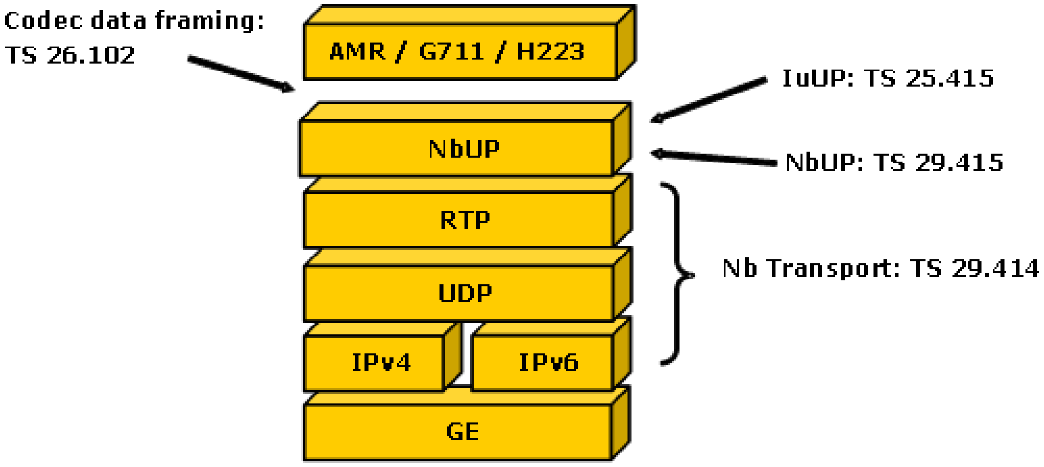
Fusion NbUP Endpoint Usage Scenario

The following illustration shows how a platform offering media services (such as ringback tones, IVR, and IVVR) can be connected in a 3GGP Release 4 Network.



Fusion NbUP Protocol Stack

The following illustration shows the protocol stack along with the relevant standards and specifications for each layer in the stack:



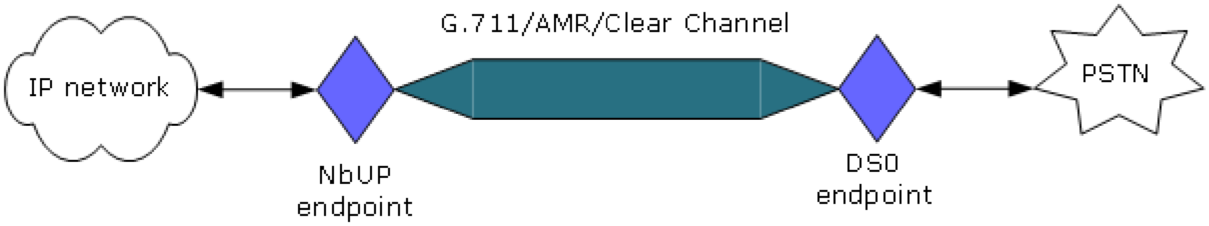
The NbUP protocol (TS 29.415) is almost identical to the Iu UP protocol (TS 25.415). Because the NbUP uses the same PDU types as the Iu UP, the term IuFP is used to refer to the common framing.

The Dialogic implementation of NbUP supports the following functions:

- Transfer of user data
- Initialization
- Error event handling
- Frame quality classification

The NbUP endpoint returns a negative acknowledgement (NACK) for all other procedures with an appropriate error code.

A typical NaturalAccess application acting as a media gateway uses an NbUP endpoint along with G.711, AMR, or clear channel connected to a DS0 endpoint to transfer media between TDM and IP interfaces as shown:



Features and Limitations

NbUP endpoint supports the following features related to NbUP protocol:

- Supports outbound (Master mode) as well as inbound (Slave mode) Iu UP INIT procedure.
- Supports PDU type 0 (with payload CRC) as well as PDU type 1 (without payload CRC) for transfer of user data.
- Supported media types include:
    - G.711/PCM – Supports 5 ms packetization and 20 ms packetization.
    - AMR – All data rates supported by Fusion are supported. (20 ms packetization time)
    - H.223 – Both 5 ms packetization and 20 ms packetization are supported.
- Frame numbering based on sent Iu UP PDU is supported. Frame numbering based on Time is not supported.
- The transport layer for Nb UP shall be RTP/IPv4. *IPv6 is not supported.*

Configuring Fusion NbUP Endpoint

Complete the following steps to instantiate and use an NbUP endpoint for audio only or 3G-324M over IP applications:

1. Create the NbUP endpoint using **mspCreateEndpoint**. After this step, the endpoint is created and all RTP related parameters are configured. No NbUP packets can be sent or received at this state.
2. Configure the NbUP protocol parameters using **mspSendCommand**. After this step, the NbUP endpoint is ready to start NbUP initialization.
3. Start the NbUP initialization using **mspSendCommand**. This step is only necessary if the endpoint was not configured to start automatically.
4. Completion of the NbUP initialization along with some other mid-session events are indicated as MSPP unsolicited events.

Create the NbUP endpoint using mspCreateEndpoint

NbUP endpoints are special purpose RTP endpoints and can use the same parameters to create NbUP endpoints. Set the endpoint type in the MSP_ENDPOINT_ADDR structure to MSP_ENDPOINT_RTPFDX_NBUP instead of MSP_ENDPOINT_RTPFDX.

For more information, refer to the *Dialogic® NaturalAccess™ Media Stream Protocol Processing API Developer's Manual*.

Configure the NbUP protocol parameters using mspSendCommand

Use the **mspSendCommand** function to configure NbUP endpoints. Use the MSP_CMD_RTPFDX_NBUP_CONFIG command code and the parameters are sent through msp_ENDPOINT_RTPFDX_NBUP_CONFIG structure, as shown:

```
typedef struct {
    U32       nbupmodemask;       //The working mode of NBUP

    #define NBUP_AUTO_START          0x1   // Bit 1 (1 – Start immediately, 0 – Don't Start)
    #define NBUP_INITMODE_MASTER     0x2   // Bit 2 (1 – Master Mode, 0 – Slave Mode)
    #define NBUP_NO_ERROR_DETECTION  0x4   // Bit 3 (1 – PDU type 1, 0 – PDU Type 0)

    U32       inittimerduration;   // initialization phase duration
    U32       initnbretry;         // number of retry upon initialization failure
    U32   codectype;               // codec type

    #define NBUP_CODEC_PCMU       10     // G.711 Mu-law
    #define NBUP_CODEC_PCMA       11     // G.711 A-Law
    #define NBUP_CODEC_AMR_475    32     // AMR 4.75 kbps
    #define NBUP_CODEC_AMR_515    33     // AMR 5.15 kbps
    #define NBUP_CODEC_AMR_59     34     // AMR 5.90 kbps
    #define NBUP_CODEC_AMR_67     35     // AMR 6.70 kbps
    #define NBUP_CODEC_AMR_74     36     // AMR 7.40 kbps
    #define NBUP_CODEC_AMR_795    37     // AMR 7.95 kbps
    #define NBUP_CODEC_AMR_102    38     // AMR 10.2 kbps
    #define NBUP_CODEC_AMR_122    39     // AMR 12.2 kbps
    #define NBUP_CODEC_H223       40     // H.223 Media

    U32   payloadid;                     // payload id for NbUP packets
    U32       frameduration;             //The frame duration (5msec or 20msec)

    #define NBUP_FD_5MSEC         5       //NbUp Frane Duration 5msec
    #define NBUP_FD_20MSEC     20    //NbUp Frane Duration 20msec

} msp_ENDPOINT_RTPFDX_NBUP_CONFIG;
```

```
msp_ENDPOINT_RTPFDX_NBUP_CONFIG nbupcfg = {0};
nbupcfg.nbupmodemask = NBUP_INITMODE_MASTER|NBUP_NO_ERROR_DETECTION;
nbupcfg.inittimerduration = 5000; // 5 secs
nbupcfg.init_nbretry = 3; // 3 retries
nbupcfg.codectype = NBUP_CODEC_PCMU; // G.711 MU-LAW
nbupcfg.payloadid = 96;
nbupcfg.frameduration = NBUP_FD_5MSEC; // 5ms packets

if ( (ret = mspSendCommand( hRtpRtcpFdxMspHd,
                            mspBuildCommand(MSP_ENDPOINT_RTPFDX_NBUP,
                            MSP_CMD_RTPFDX_NBUP_CONFIG ),
                            &nbupcfg,
                            sizeof(msp_ENDPOINT_RTPFDX_NBUP_CONFIG )) != SUCCESS ))
{
      printf("mspSendCommand Api failed for Configuring NBUP..error 0x%x\n", ret);
}
```

msp_ENDPOINT_RTPFDX_NBUP_CONFIG

The following table describes the msp_ENDPOINT_RTPFDX_NBUP_CONFIG structure:

Field	Default	Description
nbupmodemask	0x0	Bit Mask indicating the working mode for NbUP endpoint. Each bit enables/disables a particular feature as given below: Bit 1 – Auto Start (0 – Don't Start, 1 – Start Immediately) Bit 2 – Master / Slave Mode (0 – Slave Mode, 1 – Master Mode) Bit 3 – NbUP PDU type used for media transfer (0 – PDU Type 0; with Payload CRC, 1 – PDU Type 1, without Payload CRC) Bit 4 – 32 - Reserved
inittimerduration	1000 ms	Initialization phase timer value. In master mode, the endpoint waits for this period to receive a positive or negative acknowledgement (ACK/NACK) before re-transmitting the INIT message. In slave mode, the endpoint waits for inittimerduration*(initnbretry + 1) to receive an INIT message from remote end before timing out.
initnbretry	3	In master mode, the endpoint attempts these many retries in case of initialization timeout before ending the session.
codectype	10	Type of media to transfer. Also indicates the payload ID to pass onto the decoder in the connected MSPP channel.
payloadid	96	Payload ID of the RTP packets to be sent and received. These packets contain NbUP data as their payload. On the sending side, the endpoint replaces the payload ID set by the encoder in the connected MSPP channel, with this value.
frameduration	5 ms	Duration of the media put into each NbUP / RTP packet. For G.711 or H.223, the duration can be configured as either 5 ms (40 bytes payload) or 20 ms (160 bytes payload). For AMR, the duration is always 20 ms. This field also overrides the value of RTP **framequota**.

Start the NbUP initialization using mspSendCommand

Use the **mspSendCommand** function with command code MSP_CMD_RTPFDX_NBUP_START to start the NbUP Initialization phase, unless the endpoint was configured to start automatically after configuration through NBUP_AUTO_START.

```
if (ret = mspSendCommand( hRtpRtcpFdxMspHd,
mspBuildCommand(MSP_ENDPOINT_RTPFDX_NBUP,       MSP_CMD_RTPFDX_NBUP_START ),
                                      NULL, 0) != SUCCESS )
{
             printf("could not start NBUP negotiation - error 0x%x\n", ret);
}
```

Stopping the NbUP Session

Use the MSP_CMD_RTPFDX_NBUP_STOP command to stop the current NbUP session. Any session must be stopped to re-configure the endpoint with a new set of NbUP parameters.

```
if (ret = mspSendCommand( hRtpRtcpFdxMspHd,
mspBuildCommand(MSP_ENDPOINT_RTPFDX_NBUP,       MSP_CMD_RTPFDX_NBUP_STOP ),
                                      NULL, 0) != SUCCESS )
{
             printf("could not stop NBUP  session- error 0x%x\n", ret);
}
```

MSPP Unsolicited Events

The following table lists the unsolicited events returned by NbUP filters:

Event	Description
MSPEVN_NBUP_INIT	Indicates the completion of Initialization phase. Could be positive acknowledgement (ACK), negative acknowledgement (NACK), or Timeout.
MSPEVN_NBUP_ERROR	Indicates an error condition during the session.

MSPEVN_NBUP_INIT event Reason Codes

The following table summarizes the possible reason codes in MSPEVN_NBUP_INIT event:

Reason code	Description
NBUP_INIT_SUCCESS	SUCCESS
NBUP_INIT_TIMEOUT	Timeout waiting for INIT message.
NBUP_INIT_ACK_TIMEOUT	Timeout waiting for acknowledgement
NBUP_INIT_NACK	Negative acknowledgement received.

MSPEVN_NBUP_ERROR Event Reason Codes

Reason code	Description
NBUP_ERROR_CRC_HEADER	CRC error of frame header
NBUP_ERROR_CRC_PAYLOAD	CRC error of frame payload
NBUP_ERROR_UNEXP_FRAME_NB	Unexpected frame number
NBUP_ERROR_FRAME_LOSS	Frame loss
NBUP_ERROR_UNKNOWN_PDU	PDU type unknown
NBUP_ERROR_UNKNOWN_PROC	Unknown procedure
NBUP_ERROR_UNKNOWN_RSV	Unknown reserved value
NBUP_ERROR_UNKNOWN_FIELD	Unknown field
NBUP_ERROR_FRAMME_TOO_SHORT	Frame too short
NBUP_ERROR_MISSING_FIELD	Missing fields
NBUP_ERROR_UNEXP_PDU	Unexpected PDU type
NBUP_ERROR_UNEXP_PROC	Unexpected procedure
NBUP_ERROR_UNEXP_RFCI	Unexpected RFCI
NBUP_ERROR_UNEXP_VALUE	Unexpected value
NBUP_ERROR_INIT_FAILURE	Initialization failure
NBUP_ERROR_INIT_NET_FAILURE	Initialization failure (network error, timer expiry)
NBUP_ERROR_INIT_NACK_FAILURE	Initialization failure (IuUP function error, repeated negative acknowledgement)
NBUP_ERROR_RATE_CTRL_FAILURE	Rate control failure
NBUP_ERROR_EVENT_FAILURE	Error event failure
NBUP_ERROR_TIME_ALIGNMENT_NOT_SUPP	Time Alignment not supported
NBUP_ERROR_TIME_ALIGNMENT_NOT_POSS	Requested Time Alignment not possible
NBUP_ERROR_MODE_VERSION_NOT_SUPP	IuUP Mode version not supported

MSPP Unsolicited Event Structures

Fusion NbUP Endpoint uses two MSPP unsolicited event structures:

- MSPEVN_NBUP_INIT
- MSPEVN_NBUP_ERROR

MSPEVN_NBUP_INIT

```
typedef struct tag_msp_NBUP_INIT
{
    DWORD FilterId; /* NbUP Filter ID (MSP_ENDPOINT_RTPFDX_NBUP)*/
    DWORD status;   /* Success or error condition */
} msp_NBUP_INIT;
```

MSPEVN_NBUP_ERROR

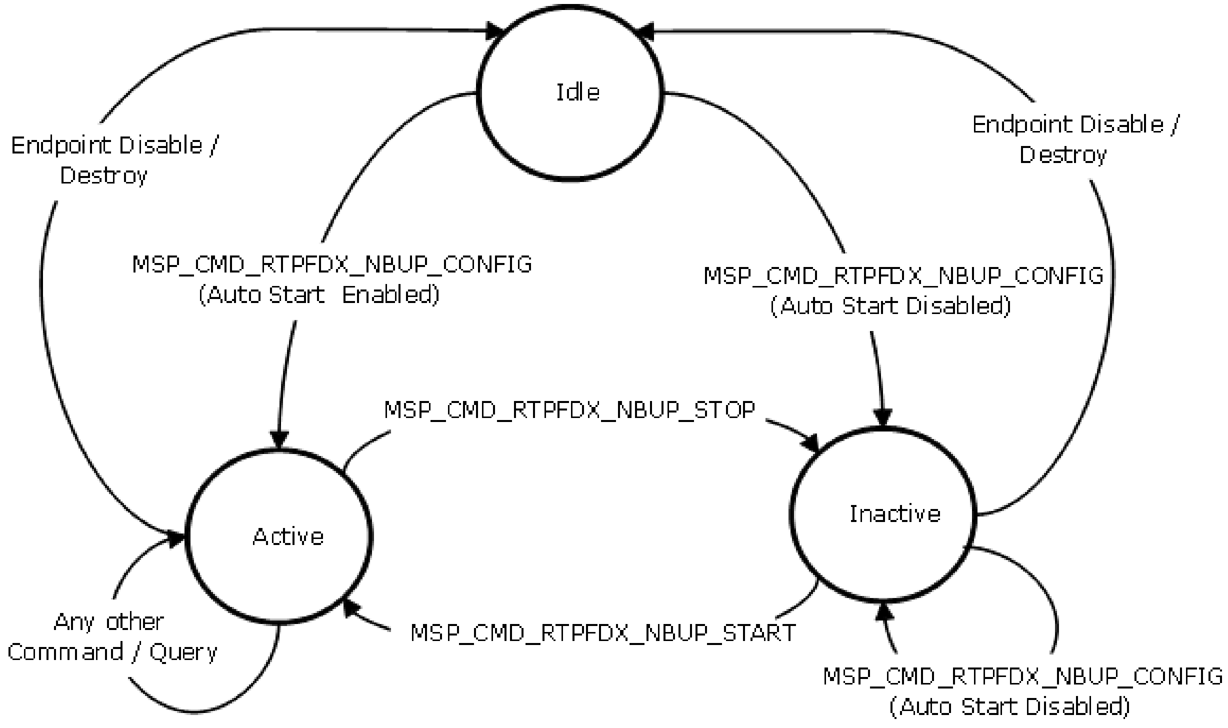
```
typedef struct tag_msp_NBUP_ERROR
{
    DWORD FilterId; /* NbUP Filter ID (MSP_ENDPOINT_RTPFDX_NBUP)*/
    DWORD error;    /* Error condition */
} msp_NBUP_ERROR;
```

MSPP State Machine

For more information, refer to the *Dialogic® NaturalAccess™ Media Stream Protocol Processing API Developer's Manual*.

NbUP State Machine

The following illustration shows the NbUP state machine and all the possible transitions:


Idle
Endpoint Disable / Destroy
Endpoint Disable / Destroy
MSP_CMD_RTPFDX_NBUP_CONFIG (Auto Start Enabled)
MSP_CMD_RTPFDX_NBUP_CONFIG (Auto Start Disabled)
MSP_CMD_RTPFDX_NBUP_STOP
Active
Inactive
Any other Command / Query
MSP_CMD_RTPFDX_NBUP_START
MSP_CMD_RTPFDX_NBUP_CONFIG (Auto Start Disabled)

Note: NbUP states and MSPP states are independent of each other.

NbUP Endpoint Status Query

An application can query the status of the endpoint when it is in Active State (both the RTP and NbUP states must be in Active State). The query returns filter status information in a buffer. For more information, refer to the *Dialogic® NaturalAccess™ Media Stream Protocol Processing API Developer's Manual*.

The buffer received with the query response is of type msp_ENDPOINT_RTPFDX_NBUP_STATUS:

```
typedef struct tag_msp_ENDPOINT_RTPFDX_STATUS {
msp_ENDPOINT_RTPFDX_STATUS rtpfdxstats;  // RTP status
msp_ENDPOINT_NBUP_STATUS          nbupstats;    // NbUP status
} msp_ENDPOINT_RTPFDX_NBUP_STATUS;
```

```
typedef struct tag_msp_ENDPOINT_NBUP_STATUS {
    DWORD         mode;
    DWORD         state;
    DWORD         rtptype;
    DWORD         rxPduFrame;
    DWORD         rxCtrlFrame;
    DWORD         txPduFrame;
    DWORD         txCtrlFrame;
    DWORD         rateCtrlError;
    DWORD         timeAlignmentError;
    DWORD         frameLossError;
    DWORD         unexpectedFrameError;
    DWORD         unexpectedPduError;
    DWORD         unknownProcError;
    DWORD         unknownPduError;
    DWORD         crcHeaderError;
    DWORD         crcPayloadError;
} msp_ENDPOINT_NBUP_STATUS;
```

For more information about the msp_ENDPOINT_RTPFDX_STATUS structure, refer to the *Dialogic® NaturalAccess™ Media Stream Protocol Processing API Developer's Manual*.

The following NbUP-related information is returned as part of msp_ENDPOINT_NBUP_STATUS:

Field	Description
mode	NbUP Mode configured.
state	Indicates the current NbUP state of the filter: Idle (0x00) The filter is created. Inactive (0x01) The filter is configured with its NbUP parameters. Active (0x02) The filter is transferring media.

Field	Description
rtptype	Type of Media being transferred. Valid values: 1 = RTP_DATA_TYPE_G711_5MS 2 = RTP_DATA_TYPE_G711_20MS 3 = RTP_DATA_TYPE_AMR 4 = RTP_DATA_TYPE_H223_5MS 5 = RTP_DATA_TYPE_H223_20MS
rxPduFrame	Received NbUP PDU message count.
rxCtrlFrame	Received NbUP Control message count.
txPduFrame	Transmitted NbUP PDU message count.
txCtrlFrame	Transmitted NbUP Control message count.
rateCtrlError	Rate control failure count.
timeAlignmentError	Time Alignment Error count.
frameLossError	Number of frames lost.
unexpectedFrameError	Unexpected frame error count.
unexpectedPduError	Unexpected PDU error count.
unknownProcError	Unexpected Control Procedure count.
unknownPduError	Unknown PDU error count.
crcHeaderError	Header CRC checksum error count.
crcPayloadError	Payload CRC checksum error count.

3. Jitter Filter

Jitter Filter RFD Command

Jitter filter supports a command for changing remote frame duration (RFD) in runtime along with existing commands.

RFD defines the duration of voice and data in each RTP packet received. Jitter filter needs to correctly assemble the received packets to match the frame size expected by the decoder. For example, the frame size expected by G.711 decoder is 10 ms. If the RTP endpoint connected to the jitter is receiving 5 ms packets, then jitter RFD needs to be defined as 5 ms. The jitter can then merge the two consecutive packets properly before forwarding it to the decoder.

The following table describes the jitter command:

Command ID	**Description**	**Units**	**Default**
MSP_CMD_JITTER_CHG_RFD	Changes jitter RFD.	ms	N/A

Use the msp_FILTER_JITTER_CMD structure to send the command to the jitter filter. For more information, refer to the *Dialogic® NaturalAccess™ Media Stream Protocol Processing API Developer's Manual*.

The following table defines the jitter RFD values:

Value	**Description**
MSP_RFD_20MS	The received packets have 20 ms frames. In this case, jitter filter forwards the frame as is.
MSP_RFD_5MS	The received packets have 5 ms frames. In this case, jitter merges frames and then forwards it to decoder.

4. Clear Channel

Clear Channel Overview

An MSPP channel is made up of one or more filters that transform a real-time flow of voice or fax data from one form to another. The application creates the MSPP channels independently of MSPP endpoints and then uses channels to connect pairs of compatible endpoints. Clear channel is a new channel type for transparently passing any data between two endpoints.

The following table provides an overview of the information provided with **mspCreateChannel** for creating a clear channel:

Channel type	Address structure	Parameter structure
Clear	Board number Channel type Enable or disable special filter attributes (with the FilterAttribs parameter)	Jitter parameters Voice encoder parameters Voice decoder parameters

Creating a Clear Channel

To create an MSPP clear channel, use MSP_CHANNEL_ADDR and any channel-specific parameter structures to specify parameters.

For more information, refer to the *Dialogic® NaturalAccess™ Media Stream Protocol Processing API Developer's Manual*.

Specify the channelType field in the MSP_CHANNEL_ADDR structure with one of the following values:

Value	Description
ClearFullDuplex	Full duplex transparent data flow between two endpoints.
ClearEncodeSimplex	Simplex transparent data flow from the Left endpoint (usually DS0) to Right endpoint (usually RTP).
ClearDecodeSimplex	Simplex transparent data flow from the Left endpoint (usually RTP) to Right endpoint (usually DS0).

Encoder Parameters

The following parameter subset is supported by the clear channel encoder component defined in the msp_FILTER_ENCODER_PARMS structure:

Field	Type	Default	Units	Description
size	DWORD	N/A	N/A	Size of the structure.
mode	WORD	0x0001 (online)	N/A	Vocoding mode. Off-line means that the encoder produces null media packet frames. 0x0000 = off-line 0x0001 = online

All other parameters are ignored by the encoder.

For more information, refer to the *Dialogic® NaturalAccess™ Media Stream Protocol Processing API Developer's Manual*.

Decoder Parameters

The clear channel decoder component defined in the msp_FILTER_DECODER_PARMS structure supports only the following subset of parameters:

Field	Type	Default	Units	Description
size	DWORD	N/A	N/A	Size of the structure.
mode	WORD	0x0001 (online)	N/A	Vocoding mode. Off-line means that the decoder filter plays silence to the DS0 endpoint. 0x0000 = off-line 0x0001 = online

All other parameters are ignored by the decoder.

For more information, refer to the *Dialogic® NaturalAccess™ Media Stream Protocol Processing API Developer's Manual*.

Configuring the Board

Clear channel needs RAW DSP resources to be configured on the board to be able to run. RAW DSP resources are built using “f_raw” DPM which has to be run in a DSP pool configured to run with no companding (NO_LAW).

When configuring a CG board for 3G-324M over IP applications, you must configure MUX and RAW DSP pools. Refer to the DSP Resource management section of CG board installation and developer's manual for general information about configuring the board.

Sample Resource Definitions

```
# ---------------------------------------------------------
# Set up the voice processing DSP's in A_LAW (for E1)
# Set up the MUX and RAW DSP's in NO_LAW so they won't compand
# ---------------------------------------------------------
DSP.C5x[0..95].XLaw                        = A_LAW
DSP.C5x[0..11].XLaw                       = NO_LAW
DSP.C5x[12..17].XLaw          = NO_LAW
# ----------------------------------------------------------------
# Very important for MUX DSP's in 3G-324M Interface configuration!
# ----------------------------------------------------------------
DSP.C5x[0..11].DataInQSize                 = 0x800
DSP.C5x[0..11].DspOutQStart              = 0x2900
DSP.C5x[0..11].DspOutQSize                = 0x900
################################################################
# RESOURCE MANAGEMENT
################################################################
################################################################
# Resource Pool 1 - MUX
################################################################
Resource[0].Name           = MUX_DEMUX
Resource[0].TCPs              = nocc
Resource[0].DSPs             = 0 1 2 3 4 5 6 7 8 9 10 11
Resource[0].Size               = 120
Resource[0].StartTimeSlot     = 0
Resource[0].Definitions          = (mux.mux & mux.demux)
################################################################
# Resource Pool 2 - RAW
################################################################
Resource[1].Name            = RAW
Resource[1].TCPs              = nocc
Resource[1].DSPs             = 12 13 14 15 16 17
Resource[1].Size               = 120
Resource[1].StartTimeSlot   = 120
Resource[1].Definitions        = (f_raw.cod & f_raw.dec)
################################################################
```

5. Demonstration Program

Changes to msppsamp

msppsamp is a demonstration program provided with NaturalAccess Fusion that demonstrates how to set up MSPP endpoints and channels for data transfer between PSTN (DS0) and IP (RTP) endpoints. For more information on *msppsamp*, refer to the *Dialogic® NaturalAccess™ Fusion™ VoIP API Developer's Manual*.

msppsamp supports the following functionality:

- Creating, configuring, and querying an NbUP endpoint.
- Clear channel.
- Streaming the NbUP data stream to another IP address using RTP switching channel for validation and capture purposes.

Command Line Switches

The following table describes the command line switches that were added or modified:

Entry	Description	Valid values
-a	Create NbUP endpoint in place of normal RTP endpoint.	N/A
-s ***destinationIPaddress***	IPv4 address of remote machine to which the NbUP stream is forwarded using RTP switching channel.	IPv4 address.
-g vocodertype	Type of encoder/decoder filter to use.	CLR and clr along with already supported values.

RTP Endpoint Filter Commands

The following commands were added to the RTP endpoint filter commands:

Option	Sub-option	Description
B	RTP endpoint filter commands.	
	7	Configure NbUP protocol parameters.
	8	Start NbUP initialization procedure.
	9	Stop NbUP initialization procedure.

Jitter Filter Commands

The following commands were added to the Jitter filter commands:

Option	Sub-option	Description
C	Jitter filter commands	
	4	Change RFD.