BRI/2 User's Guide

For Windows

Copyright © 1999-2000 Dialogic Corporation

05-1466-001

COPYRIGHT NOTICE

Copyright © 1999-2000 Dialogic Corporation. All Rights Reserved.

All contents of this document are subject to change without notice and do not represent a commitment on the part of Dialogic Corporation. Every effort is made to ensure the accuracy of this information. However, due to ongoing product improvements and revisions, Dialogic Corporation cannot guarantee the accuracy of this material, nor can it accept responsibility for errors or omissions. No warranties of any nature are extended by the information contained in these copyrighted materials. Use or implementation of any one of the concepts, applications, or ideas described in this document 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 condone or encourage such infringement. Dialogic makes no warranty with respect to such infringement, nor does Dialogic waive any of its own intellectual property rights which may cover systems implementing one or more of the ideas contained herein. Procurement of appropriate intellectual property rights and licenses is solely the responsibility of the system implementer. The software referred to in this document is provided under a Software License Agreement. Refer to the Software License Agreement for complete details governing the use of the software.

All names, products, and services mentioned herein are the trademarks or registered trademarks of their respective organizations and are the sole property of their respective owners. DIALOGIC (including the Dialogic logo), DTI/124, and SpringBoard are registered trademarks of Dialogic Corporation. A detailed trademark listing can be found at: http://www.dialogic.com/legal.htm.

Publication Date: January, 2000

Part Number: **Error! No text of specified style in document.**

Dialogic, an Intel Company
1515 Route 10
Parsippany NJ 07054
U.S.A.

For **Technical Support**, visit the Dialogic support website at:
http://support.dialogic.com

For **Sales Offices** and other contact information, visit the main Dialogic website at:
http://www.dialogic.com

OPERATING SYSTEM SUPPORT

The term *Windows* refers to both the Windows NT® and Windows® 2000 operating systems. For a complete list of supported Windows operating systems, refer to the *Release Guide* that came with your Dialogic System Release for Windows, or to the Dialogic support site at *http://support.dialogic.com/releases*.

Table of Contents

1. Introduction 1
1.1. What's In This Guide 1
1.2. Supported Hardware 1
1.3. Voice Features Supported 1
1.4. ISDN Features Supported 5
1.5. ISDN Protocols 8
1.6. Fax Features Supported 9
1.7. Data Features Supported 9
1.8. Related Publications 9
2. The ISDN Interface 11
2.1. B and D Channel Signaling 12
2.1.1. Physical Layer 12
2.1.2. Data Link Layer (D-channel) 13
2.1.3. Network Layer (D-channel) 14
3. Voice Features 21
4. Dialogic DSP-based Group 3 Fax Features 23
4.1. Overview of DSP-based Group 3 Fax 23
5. Data Features 25
5.1. NDIS Compatibility 25
5.2. Remote Access Service (RAS) 28
5.3. PPP and PPP-MP 28
5.4. An Example Using Data Transfer 29
5.4.1. PBX Setup 29
5.4.2. Computer Setup 30
5.5. The Dialogic BRI/2 Configuration Utility 39

List of Tables

Table 1. Analog Call Control Functions Not Supported 1
Table 2. Supported Springware Functions 2
Table 3. ISDN Function Supported 6
Table 4. ETSI Specification Cross-Reference for Supplementary Services 19

List of Figures

Figure 1. BRI/2 Interfaces 11
Figure 2. BRI Layer 1 Frame 13
Figure 3. BRI Layer 2 Frame (D Channel) 13
Figure 4. BRI Layer 3 Frame (D Channel) 14
Figure 5. Information Element Format 18
Figure 6. NDIS Driver Model 26
Figure 7. Data Transfer Model 27
Figure 8. Data Transfer Test Setup 30

1. Introduction

1.1. What's In This Guide

This User's Guide contains information specific to how ISDN is implemented on the BRI/2 board. For information about installing and configuring BRI/2 hardware, refer to the *Installation and Configuration Guide* that came with your Dialogic System Release Software.

1.2. Supported Hardware

This package supports the Dialogic BRI/2 board. The BRI/2 is a PCI board with two BRI S_0 interfaces. The BRI/2 emulates two standard BRI station sets with display, and is designed to support the Euro-ISDN protocol. The BRI/2 provides all of the voice processing functionality of the existing Springware™ -based products, plus the ability to support many enhanced ISDN features. In addition, the BRI/2 can facilitate four instances of Dialogic DSP-based Group 3 Fax and provides ISDN B channel data communications.

Voice and Dialogic DSP-based Group 3 Fax are supported by the Dialogic Windows driver, and data communications is supported through applications that support the Windows NDIS driver.

1.3. Voice Features Supported

The BRI/2 uses Springware™ voice functions to perform analog voice processing. Since the BRI/2 also uses the Euro-ISDN protocol, which is a digital protocol, the following analog functions are not supported (the equivalent ISDN functions are also listed):

Table 1. Analog Call Control Functions Not Supported

Analog Call Control Functions	ISDN Call Control Functions
dx_open( ) – opens a voice channel	**cc_Open( )** – opens an ISDN call control

with an associated loop start interface.	device.
dx_close( ) – closes a voice channel with an associated loop start interface.	**cc_Close( )** – closes an ISDN call control device.
dx_dial( ) – sends digits (DTMFs or MFs) to the network to make a call.	**cc_MakeCall( )** – makes a call on an ISDN line (sends a SETUP message to the network).
dx_wtring( ) – Synchronously waits for an inbound ring event.	**cc_WaitCall( )** – enables an ISDN call control device to process an incoming call
dx_setevtmsk( ) – enables detection of asynchronous ring events.	**cc_WaitCall( )** – enables an ISDN call control device to process an incoming call.
dx_sethook( ) – used to place a loop start interface offhook to answer or make a call.	**cc_AcceptCall( )** – indicates to the remote end that the ISDN call control device is "ringing". **cc_AnswerCall( )** – accepts a connection request on an ISDN call control device.
dx_sethook( ) – used to place a loop start interface onhook to disconnect a call.	**cc_DropCall( )** – Disconnects a call on an ISDN call control device. **cc_ReleaseCall( )** – Releases all driver and firmware resources associated with a disconnected call on an ISDN call control device.

An existing application written for any analog Dialogic voice processing product may be used with the BRI/2, however the call control (call processing) portion of the application will require changes. The following analog voice functions are supported on the BRI/2. Refer to the *Voice Software Reference: Programmer's Guide* for information on using Voice Functions.

Table 2. Supported Springware Functions

Function	Description
ATDX_ANSRSIZ( )	returns the duration of the answer

Function	Description
ATDX_BDNAMEP( )	returns a pointer
ATDX_BDTYPE( )	returns the device type
ATDX_BUFDIGS( )	returns the number of uncollected digits
ATDX_CHNAMES( )	returns a pointer to an array
ATDX_CHNUM( )	returns the channel number
ATDX_CONNTYPE( )	returns the connection
ATDX_CPTERM( )	returns last Call Analysis termination
ATDX_CRTNID( )	returns the tone identifier
ATDX_DTNFAIL( )	returns character for dial tone
ATDX_FRQDUR( )	can be used to return the duration of a SIT tone
ATDX_FRQDUR2( )	can be used to return the duration of a SIT tone
ATDX_FRQDUR3( )	can be used to return the duration of a SIT tone
ATDX_FRQHZ( )	return frequency of answered signal
ATDX_FRQHZ2( )	return frequency of second detected tone
ATDX_FRQHZ3( )	return frequency of third detected tone
ATDX_FRQOUT( )	returns percentage of a single tone frequency
ATDX_HOOKST( )	returns the current hook state
ATDX_LINEST( )	returns a bitmapped representation of activity
ATDX_LONGLOW( )	returns duration of the longer silence
ATDX_PHYADDR( )	returns the physical address
ATDX_SHORTLOW( )	returns duration of shorter silence
ATDX_SIZEHI( )	returns duration of initial non
ATDX_TERMMSK( )	returns a bitmap
ATDX_TONEID( )	returns the user
ATDX_TRCOUNT( )	returns number of bytes transferred
dx_addspddig( )	sets a DTMF digit to adjust speed
dx_addtone( )	adds the tone

Function	Description
dx_addvoldig( )	sets a DTMF digit to immediately adjust volume
dx_adjsv( )	adjusts speed or volume
dx_blddt( )	defines a simple dual frequency tone
dx_blddtcad( )	defines a simple dual frequency cadence tone
dx_bldst( )	defines a simple single frequency tone
dx_bldstcad( )	defines a simple single frequency cadence tone
dx_bldtngen( )	sets up tone generation template
dx_chgdur( )	alters standard definition of duration component
dx_chgfreq( )	changes the standard definition
dx_chgrepcnt( )	changes the standard definition
dx_clrcap( )	clears all the fields in a DX_CAP structure
dx_clrdigbuf( )	causes the digits present in the firmware digit buffer
dx_clrsvcond( )	clears any speed or volume adjustment conditions
dx_clrtpt( )	clears all DV_TPT fields
dx_deltones( )	removes all user
dx_dial( )	dials an ASCIIZ string
dx_distone( )	disables detection of TONE ON
dx_enbtone( )	enables detection of TONE ON
dx_fileclose( )	closes the file associated with the handle
dx_fileopen( )	opens the file specified by filep
dx_fileread( )	turns number of bytes read by application.
dx_filelseek( )	moves file pointer associated with handle
dx_filewrite( )	writes count byes from buffer into file associated with handle
dx_getcursv( )	returns the specified channel's current speed
dx_getdig( )	initiates the collection of digits
dx_getevt( )	used to synchronously monitor channels
dx_getsvmt( )	returns contents of Speed or Volume Modification Table

Function	Description
dx_initcallp( )	initializes and activates PerfectCall Call Analysis
dx_play( )	plays recorded voice data
dx_playf( )	synchronously plays voice data
dx_playiottdata( )	plays back recorded voice data from multiple sources
dx_playtone( )	plays tone defined by TN_GEN template
dx_playvox( )	plays voice data stored in a single VOX file
dx_playwav( )	plays voice data stored in a single WAVE file
dx_rec( )	records voice data from a single channel
dx_recf( )	permits voice data to be recorded
dx_reciottdata( )	records voice data to multiple destinations
dx_recvox( )	records voice data to a single VOX file
dx_recwav( )	records voice data to a single WAVE file
dx_setdigbuf( )	sets the digit buffering mode
dx_setdigtyp( )	controls the types of digits
dx_setevtmsk( )	enables detection of Call Status Transition (CST) event
dx_setgtdamp( )	sets up the amplitudes
dx_setsvcond( )	sets adjustments and adjustment conditions
dx_setsvmt( )	updates the speed or volume
dx_setuio( )	allows an application to install a user I/O routine
dx_stopch( )	forces termination of currently active I/O functions
r2_creatfsig( )	defines and enables leading edge detection
r2_playbsig( )	plays a specified backward R2MF signal

1.4. ISDN Features Supported

The BRI/2 provides applications access to the many features and benefits of ISDN using the **Basic Rate Interface (BRI)**, which allows the transfer of both voice and data over standard 64 kbps lines.

The following ISDN functions are supported on the BRI/2. Refer to the *ISDN Software Reference* for information on using ISDN functions.

Table 3. ISDN Function Supported

API Functions	Description
cc_AcceptCall( )	responds to an incoming call request
cc_AnswerCall( )	accepts a connection request from the remote end
cc_CallAck( )	send the first response to an incoming call
cc_CallProgress( )	sends a "Progress" message to the network
cc_CallState( )	retrieves the state of a call
cc_CauseValue( )	retrieves the error/cause code of a failure
cc_Close( )	closes a previously opened line device
cc_CRN2LineDev( )	returns the line device number
cc_DropCall( )	allows the application to disconnect a call
cc_GetANI( )	retrieves Automatic Number Identification (ANI) information
cc_GetCallInfo( )	gets the information elements of the incoming message
cc_GetCRN( )	retrieves the call reference number for the event
cc_GetDLinkState( )	retrieves the logical data link state
cc_GetDNIS( )	gets the dialed number information string
cc_GetEvtMsk( )	retrieves the current ISDN event mask
cc_GetLineDev( )	retrieves the line device handle for an event
cc_GetMoreDigits( )	collects more digits via overlap receiving
cc_GetNonCallMsg( )	retrieves non Call associated ISDN messages
cc_GetParm( )	gets the current parameter values
cc_GetVer( )	retrieves the firmware version number
cc_HoldAck( )	accepts a hold request from remote equipment
cc_HoldCall( )	places an active call on hold

API Functions	Description
cc_HoldRej( )	rejects a hold request from remote equipment
cc_MakeCall( )	request a connection on the specified line device
cc_Open( )	opens a device
cc_ReleaseCall( )	release all resources
cc_ReleaseCallEx( )	release all Dialogic ISDN resources
cc_Restart( )	resets the channel to an idle state
cc_ResultMsg( )	interprets the function return code
cc_ResultValue( )	gets an error/cause code
cc_RetrieveAck( )	accept a retrieve request from remote equipment
cc_RetrieveCall( )	retrieve a call from HOLD state
cc_RetrieveRej( )	reject a retrieve request from remote equipment
cc_SetEvtMsk( )	sets the event mask
cc_SetInfoElem( )	sets additional information elements
cc_SetMinDigits( )	sets the minimum number of digits to be collected
cc_SetParm( )	sets the default channel parameters
cc_SndMsg( )	sends a call state associated ISDN message
cc_SndNonCallMsg( )	sends a non-call associated ISDN message
cc_StartTrace( )	start the capture of all D channel information
cc_StopTrace( )	stops the trace
cc_WaitCall( )	sets up conditions for processing an incoming call

The following IDSN functions listed in the *ISDN Software Reference* are **NOT** supported on the BRI/2.

- **cc_TermRegisterResponse( )**
- **cc_SndFrame( )**
- **cc_SetChanState( )**
- **cc_SetBilling( )**
- **cc_SetCallingNum( )**
- **cc_SetUsrAttr( )**
- **cc_GetSAPI( )**

- **cc_GetBChanState( )**
- **cc_GetBilling( )**
- **cc_GetFrame( )**
- **cc_GetSigInfo( )**
- **cc_GetNetCRV( )**
- **cc_GetCES( )**
- **cc_GetUsrAttr( )**
- **cc_GetDChanState( )**
- **cc_ReqANI( )**
- **cc_SetDChanCfg( )**
- **cc_GetParmEX( )**
- **cc_SetParmEX( )**
- **cc_GetInfoElem( )**

1.5. ISDN Protocols

Like any evolving technology, a single standard ISDN implementation has yet to be agreed upon worldwide. Standards have been established in a number of countries or regions. The following table lists the Euro-ISDN protocols that are currently supported on the BRI/2.

- Euro-ISDN BRI, Fax G3 and V.17 modem (software implementation of G3 Fax).
- PPP/MP protocol on B-Channels accessed via a Windows TCP/IP stack compatible NDIS driver provided by Dialogic.
- Multi-link PPP (Channel bundling)

1.6. Fax Features Supported

Dialogic DSP-based Group 3 Fax (also referred to as DSP Fax) is a software-based fax solution supported on the BRI/2 board running Dialogic System Release software. DSP Fax supports the current fax library API with some additions and modifications as described later in this chapter.

Key features of DSP Fax include:

- Four channels of voice and fax per board

- Maximum of 16 fax channels per system (4 BRI/2 boards in one system)
- Software-based fax modem
- Compatibility with ITU-T Group 3 (T.4, T.30), ETSI NET/30

Refer to the *Fax Software Reference for Windows(P/N 05-0172-007)* for information on using DSP Fax with the BRI/2.

1.7. Data Features Supported

The Dialogic drivers used with the BRI/2 enables the BRI/2 to function like a standard ISDN network interface card (NIC). Any application that uses an ISDN NIC can use the BRI/2. The NIC driver is installed by adding a network adapter using the Windows Control Panel. To enable applications to communicate with the BRI/2, the Dialogic driver implements the Windows NDIS driver.

NDIS (Network Device Interface Specification) is a Microsoft® standard that allows for multiple network adapters and multiple protocols to coexist. NDIS permits the high-level protocol components to be independent of the BRI/2 by providing a standard interface. This means that the BRI/2 may be used with applications that use the standard networking APIs that are part of the Windows operating system, such as Sockets (WinSock), NetBEUI, and SPX/IPX.

1.8. Related Publications

The following documents are available for use with this release :

- *Fax Software Reference for Windows*
- *ISDN Software Reference*
- *Voice Software Reference for Windows.*

2. The ISDN Interface

The Dialogic BRI/2 provides ISDN network access via the Basic Rate Interface (BRI). The BRI/2 also supports features such as a digital station interface, enabling direct access to BRI station sets (telephones) from PC-based computer telephony (CT) systems. The BRI/2 connects directly to a EURO-ISDN NTI or NT2 device.

The BRI/2 front-end interface complies with all ETS 300 012 and ANSI T1.605 requirements. The circuitry uses a BRI "S/T" interface, IOM-2 to PCM conversion, and HDLC serial communications. *Figure 1* shows the how the BRI/2 interface compares to the standard ISDN model.

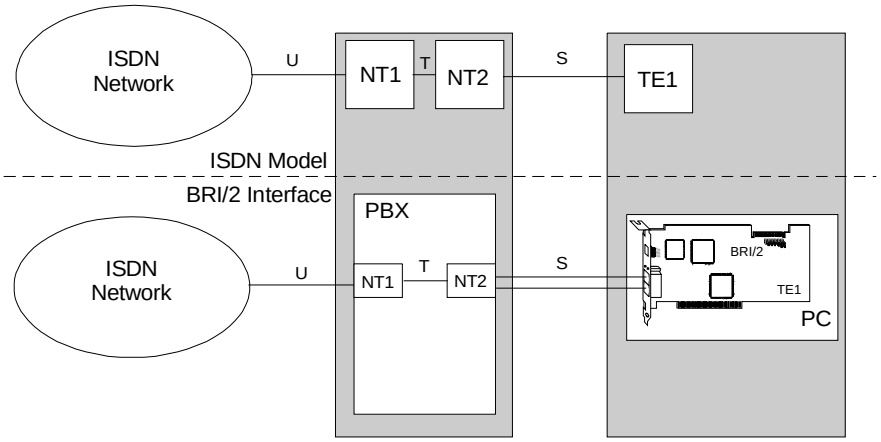


Figure 1. BRI/2 Interfaces

An NT1 device handles OSI (Open Systems Interconnection) model layer 1 functions such as the physical connection between the ISDN network and user devices. It splits the duplexed transmit and receive signals from the ISDN network (over the U interface) into separate transmit and receive components.

An NT2 device provides the physical and electrical termination for an ISDN line. It provides an interface between the twisted-pair wires used by the PBX, and the

wires used by the BRI/2. A device designated as TE1 (Terminal Equipment) is any end-user equipment that uses ISDN protocols and supports ISDN services. The interface between an NT1 device and an NT2 device is called an "T" interface.

2.1. B and D Channel Signaling

Data sent between the BRI/2 and the PBX is transmitted over the "S/T" interface. The way in which this data is transferred is controlled by protocols. Protocols are a set of rules that define the format and timing of data transmission between two devices. BRI protocols conform to the first three layers of the OSI model; physical, data link, and network.

2.1.1. Physical Layer

ITU-T Recommendation I.430 defines the physical layer specification for BRI. The physical layer consists of both the hardware (wiring, termination, etc.) and data transmission method. In BRI, two channels of user data (B channels) and one signaling channel (D channel) are transmitted over a single line (a line consists of two twisted-wire pairs; one for transmit and one for receive). To accommodate this, a technique called framing is used. Digital data is sent in frames and contains information from each of the channels, providing a "snapshot" of the data being transmitted at any given time. BRI uses common channel signaling (CCS). This means that one signaling channel (D channel) carries signaling information for more than one user data channel (B channel). Common channel signaling also allows the transmission of additional information, such as ANI and DNIS digits, over the signaling channel.

For example, in telephony applications, *user data* is usually digitally encoded voice data (PCM). In file transfer applications, *user data* is packets of HDLC encoded information. *Signaling data* carries information such as the current state of the channel (i.e., whether the telephone is on-hook or off-hook). The *signaling data* also indicates who is calling, what type of call it is (data/voice), and what number was dialed.

The figure below represents one BRI layer 1 frame. Each frame contains 48 bits.

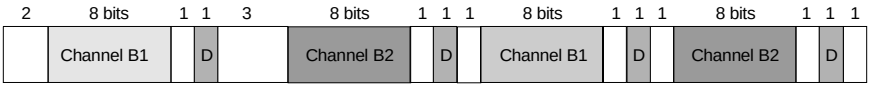


Figure 2. BRI Layer 1 Frame

2.1.2. Data Link Layer (D-channel)

The data link layer is where reliable communication between the two physically connected devices takes place. Protocols at this layer deal with setup, maintenance, and disconnection. **ITU-T Recommendation Q.920 (ETS 300 125) and Q.921 (LAPD)** define the D channel data link layer specification for BRI. LAPD transmits a stream of bits in a defined structure (frame), refer to *Figure 3. BRI Layer 2 Frame (D Channel)*. The receiver interprets the frame. The purpose of the LAPD protocol is to prepare and transmit information between layer 3 entities.

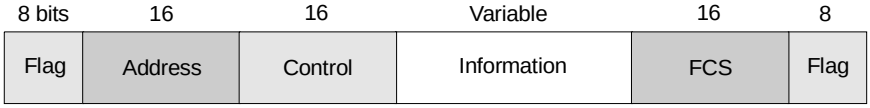


Figure 3. BRI Layer 2 Frame (D Channel)

The Control bits determine the type of frame being sent (information, supervisory, or unnumbered). Information frames are used to transfer data in sequentially numbered frames containing data from the Network layer (layer 3).

2.1.3. Network Layer (D-channel)

ITU-T Recommendation Q.931 (ETS 300 102-1) is a layer 3 ISDN protocol. This protocol deals with signaling procedures, call control, and access to and control of supplemental services. Setting up calls, providing call maintenance, and terminating the call are handled by the exchange of a series of messages between the BRI/2 and the switch. These messages are carried in LAPD frames (see *Figure 4*).

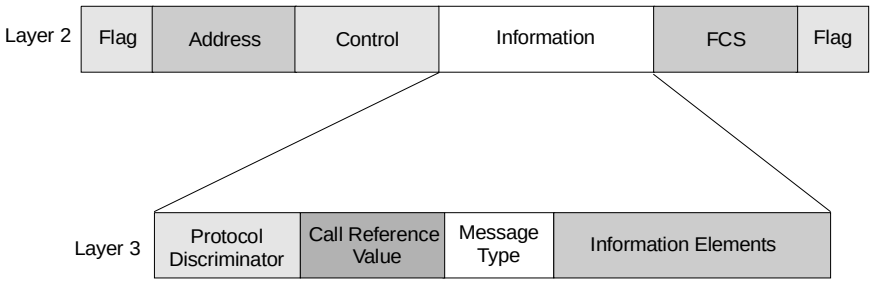


Figure 4. BRI Layer 3 Frame (D Channel)

The message format comprises variable length fields with the following general format:

- Protocol Discriminator
- Call Reference Value
- Message Type
- Information Elements

The *Protocol Discriminator* identifies the protocol type used to handle layer three messages. The *Call Reference Value* is a value assigned to a call for its duration. It has local significance (i.e., it applies to layer 3 messages which are employed between the user and network at either the originating or the terminating end). The *Message Type* is the set of messages used for establishing, controlling and tearing down a call. *Information Elements* are used with the message to provide additional information on the type and requirements of the call and are determined by the application to be supported by the connection.

Supplemental Services

Using ISDN API functions, the BRI/2 can perform the following Supplemental Services:

- Called/Calling Party Identification
- Subaddressing
- Hold/Retrieve
- Call Transfer
- Message Waiting

Call Hold and Retrieve are invoked using the following functions:

- **cc_HoldAck( )**
- **cc_HoldCall( )**
- **cc_HoldRej( )**
- **cc_RetrieveAck( )**
- **cc_RetrieveCall( )**
- **cc_RetrieveRej( )**

The other Supplementary Services are typically invoked by sending information from the BRI/2 to the PBX using an appropriate API function. This information is sent as the part of the layer 3 frame called the *Information Element* (refer to Figure 4). In order for the PBX to interpret the Information Elements as Supplementary Service requests, the Information Elements must be sent as Facility Messages.

The following functions can be used to send Facility Messages:

- **cc_SndMsg( )** - Sends a call state associated message to the PBX.
- **cc_SndNonCallMsg( )** - Sends a non-call state related messages to the PBX. This function does not require a call reference value.
- **cc_SetInfoElem( )** - Sets an information element (IE) allowing the application to include application-specific ISDN information elements in the next outgoing message.

The following functions are used to retrieve Facility Messages:

- **cc_GetCallInfo( )** - Retrieves the information elements associated with the CRN.
- **cc_GetNonCallMsg( )** - Retrieves a non-call state related ISDN messages to the PBX.

The **cc_SndMsg( )** and **cc_SndNonCallMsg( )** functions are used to send Facility Messages or Notify Messages to the PBX. The Facility Message (as defined in ETS 300-196-1) is composed of the following elements (see also *Figure 5*):

- Protocol discriminator
- Call reference
- Message type
- Facility Information Element

The Supplementary Service to be invoked and its associated parameters are specified in the Information Element. This information is PBX-specific and should be provided by the PBX manufacturer. Facility Messages are sent using the **cc_SndMsg( )** or **cc_SndNonCallMsg( )** function with msg_type = SndMsg_Facility. These functions format the Facility Message (inserting the protocol discriminator, call reference value [only for **cc_SndMsg( )**] , and message type elements), adds the Information Element data (stored in an application buffer) and sends all the information to the PBX. The PBX, in turn, interprets and acts on the information, and sends a reply to the BRI/2.

As an example, to invoke Supplementary Service 'X', you could use the **cc_SndMsg( )** function with msg_type = SndMsg_Facility. The Information Element would be defined in a data structure as follows:

```
ieblk.length = 11;
ieblk.data[0]  = 0x1c;  /* IE Identifier */
ieblk.data[1]  = 0x09;  /* Length of information */
ieblk.data[2]  = 0x91;  /* Protocol Profile */

/* information */
ieblk.data[3]  = 0xa1;  /* Component Type */
ieblk.data[4]  = 0x06;  /* Component Length */
ieblk.data[5]  = 0x02;  /* invoke tag id */
ieblk.data[6]  = 0x01;  /* invode tag length */
ieblk.data[7]  = 0x00;  /* invoke id */
ieblk.data[8]  = 0x02;  /* operation tag */
ieblk.data[9]  = 0x01;  /* operation length */
ieblk.data[10] = 0x06;  /* operation */
```

NOTE: The information included in the Information Element is determined by the Supplementary Service being invoked.

The data sent to the switch would be formatted as follows:

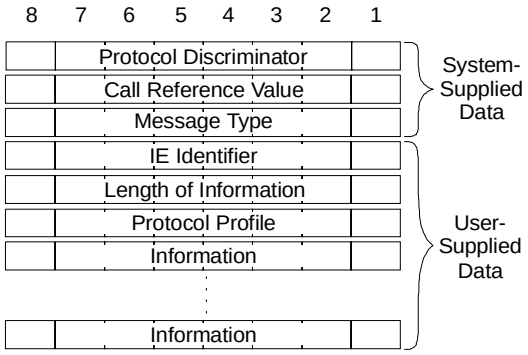


Figure 5. Information Element Format

Information elements can also be sent using the **cc_SetInfoElem( )** function, which allows the BRI/2 to send application-specific information elements in the next outgoing message. This function is used for rapid deployment of an application that "interworks" with the PBX. A typical application is user-to-user information elements in each outgoing message. This function must be used immediately before calling a function that sends an ISDN message. The information elements specified by this function are applicable only to the next outgoing ISDN message.

When a Supplementary Service is invoked, the network may return a NOTIFY Message to the user. This message can be retrieved using the **cc_GetCallInfo( )** function.

The Notify Message (as defined in ETS 300-196-1) is composed of the following elements:

- Protocol discriminator
- Call reference
- Message type
- Notification Indicator

The Notify message is coded as follows:

8	7	6	5	4	3	2	1
x	x	x	x	x	x	x	x
Protocol discriminator							
x	x	x	x	x	x	x	x
Call reference							
x	x	x	x	x	x	x	x
Message Type							
0	0	1	0	0	1	1	1
	Notification Indicator Information element identifier						
0	0	0	0	1	0	0	1
Length of Notification Indicator contents							
1/1	x	x	x	x	x	x	x
ext.	Notification Description						
0	x	x	x	x	x	x	x
ext.	Notification Description						
1	0	1	0	0	0	0	1
Notification Data Structure							

Coding requirements for other supported Supplementary Services are listed in Table 4.

Table 4. ETSI Specification Cross-Reference for Supplementary Services

Supplementary Service/Description	ETS 300 Specification
Explicit Call Transfer - enables a user (user A) to transform two of that user's calls (an active call and a held call), each of which can be an incoming call or an outgoing call, into a new call between user B and user C. "Call Transferred Alerting" and "Call Transferred Active" messages are returned by the network to the user.	367/369/369

Supplementary Service/Description	**ETS 300 Specification**
Call Hold/ Retrieve - allows a user to interrupt communications on an existing call and then subsequently, if desired, re-establish communications. When on Hold, the user may retrieve that call from hold, originate a new call, retrieve another call, or establish connection to an incoming call, e.g. a waiting call. Send a facility message with the message type data set to	139/140/141
Subaddressing (allows direct connection to individual extensions or devices sharing the same phone number, or, as a proprietary messaging mechanism). Provides additional addressing above the ISDN number of the called user.	059/060/061
Called/Calling Party Identification CLIP - Provides the calling user's ISDN number and subaddress information to the called user. This information is sent in the "Setup message" (see ETS300 102-1) by the calling user to the switch, and from the switch to the called user.	089/091/092
Called/Calling Party Identification CLIR - Restricts presentation of the calling user's ISDN number to the called user.	090/091/093
Called/Calling Party Identification COLP - Provides the calling user's ISDN number to the called user.	094/096/097
Called/Calling Party Identification COLR - restricts the ISDN and the subaddress of the called user.	095/096/098
Advice of Charge - S	178/181/182
Advice of Charge - D	179/181/182
Message Waiting Indication	650/745-1/356-20

3. Voice Features

To perform voice processing, the standard Dialogic APIs are used. This means that an existing application that was written for any Dialogic voice processing product may be used with the BRI/2. If the application was written for an analog, T1, or E1 Dialogic card, then the call control (call processing) portion of the application will require changes.

These APIs are documented in the *Voice Software Reference for Windows NT*.

To perform call processing, the call control Dialogic ISDN APIs are used. These APIs are documented in the *ISDN Software Reference for Windows*.

4. Dialogic DSP-based Group 3 Fax Features

This chapter provides an overview of Dialogic DSP-based Group 3 Fax for Windows (referred to as DSP Fax) and its system requirements. It also discusses changes to Dialogic voice and fax documentation resulting from this Fax release.

4.1. Overview of DSP-based Group 3 Fax

DSP-based Group 3 Fax is a software-based fax solution supported on the Dialogic BRI/2 board running Dialogic System Software for Windows. DSP-based Group 3 Fax supports the current fax library API with some additions and modifications as described later in this chapter.

Key features of DSP-based Group 3 Fax include:

- Four channels of voice and fax per board
- Maximum of 16 fax channels per system
- Software-based fax modem
- Compatibility with ITU-T Group 3 (T.4, T.30), ETSI NET/30.

5. Data Features

The Dialogic drivers used with the BRI/2 enables the BRI/2 to function like an ISDN network interface card (NIC). The BRI/2 provides link layer access (across the B channel) that allows reliable transfer of data across an ISDN network. To enable Windows applications to communicate with the BRI/2, the Dialogic driver implement an NDIS driver.

5.1. NDIS Compatibility

NDIS (Network Device Interface Specification) is a Microsoft® standard that allows for multiple network adapters and multiple protocols to coexist. NDIS permits the high-level protocol components to be independent of the BRI/2 by providing a standard interface. This means that the BRI/2 may be used by applications that use the standard networking APIs that are part of the Windows operating system, such as Sockets (WinSock), NetBIOS, and SPX/IPX. The NDIS Wrapper is made up of the NDIS WAN driver and the BRI/2 WAN Miniport driver. Both the LAN (transport) protocols and the TAPI protocols communicate with the NDIS WAN driver (refer to *Figure 6*).

The BRI/2 WAN Miniport driver (*dlgcbri.sys*) is installed when you install the BRI/2 ISDN Network Adapter driver.

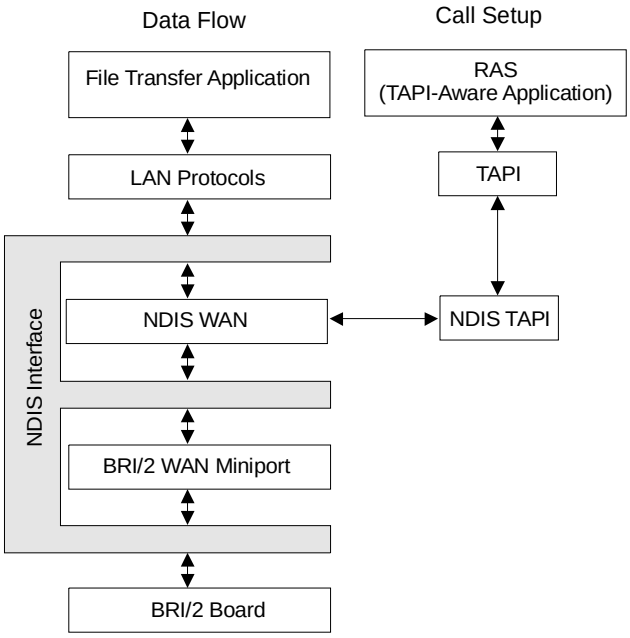


Figure 6. NDIS Driver Model

The BRI/2 WAN Miniport driver is at the bottom of the network architecture. Because the Windows network architecture supports NDIS , it requires that network adapter card drivers be written to the NDIS 3.0 specification. The BRI/2 driver is written to the NDIS specification. In Windows NT, NDIS has been implemented in a module called *ndis.sys*, which is referred to as the NDIS interface.

NDIS enables the following:

- The Dialogic driver to receive a network packet from any one of several upper layer drivers for transmission on the network.
- The Dialogic driver to accept a network packet from the BRI/2 and pass it up to one or more upper layer drivers that may receive it.
- The Remote Access Service (RAS) to set specific configuration parameters for the BRI/2.

- Upper layer drivers to query the Dialogic driver for specific configuration and statistics from the BRI/2.
- The Dialogic driver to asynchronously inform an overlying driver of changes in network status or BRI/2 status.

User data transmitted over the B channels are sent in HDLC frames. These frames are 8-bits long and are sent in the appropriate B channel block in layer 1 (refer to *Figure 7*.) The BRI/2 extracts the HDLC data and sends a raw bit stream to the NDIS driver.

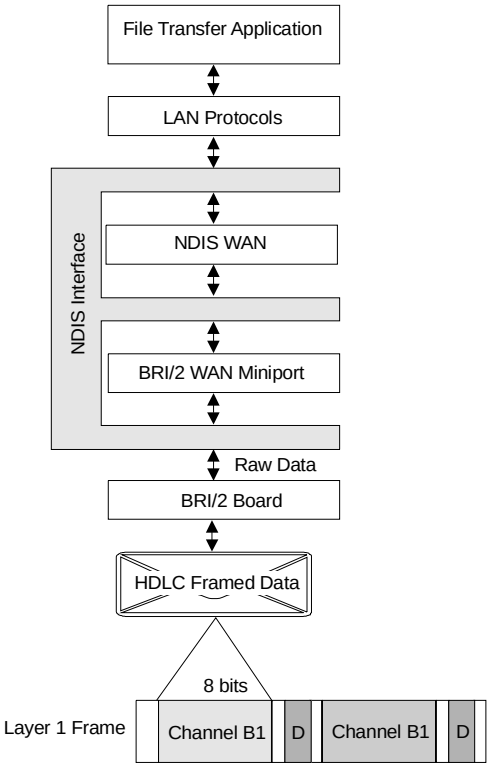


Figure 7. Data Transfer Model

When using the BRI/2 as an ISDN NIC to transfer files, the following procedures occur:

- A TAPI-aware application, such as RAS (Remote Access Service), sends a call connect request through the NDIS TAPI driver to the NDIS WAN driver. The BRI/2 Miniport Driver then translates the TAPI call control information into instructions used by the BRI/2 to place a call. TAPI is used to support essentially all dialing functions within the Windows operating system.
- After the telephone connection has been established, the RAS is used to connect remote computers for data transfer applications. Windows also supports the PPP (Point-to-Point Protocol) in RAS. The PPP is a set of framing and authentication protocols. It negotiates configuration parameters for multiple layers of the OSI model, thereby enabling any combination of IPX, TCP/IP, and NetBEUI.

5.2. Remote Access Service (RAS)

RAS is the Remote Access Service in Windows NT. RAS works in conjunction with the Dialup Networking Applet. Together with TAPI, PPP, and NDIS, the user is capable of interacting with the service selections provided by the specified dialup networking setup.

RAS performs the following:

1. Establishes framing rules between the host and the remote.
2. Initiates PPP authentication protocols to authenticate the remote.
3. Uses network control protocols enable and configure the host for the WAN protocols that will be used on the remote. The host and remote can now transfer data using any supported protocol.

5.3. PPP and PPP-MP

Point-to-Point Protocol (PPP) is a method for exchanging data packets between two computers. PPP can carry different network layer protocols over the same link. When the PPP connection sequence is successfully completed, the remote client and RAS server can begin to transfer data using any supported protocol,

such as Windows Sockets, RPC, or NetBIOS. WinSock (Windows Sockets) is an protocol that provides an interface to Windows networking protocols.

PPP Multilink provides the ability to aggregate two or more physical connections to form one larger logical connection, improving bandwidth and throughput for remote connections. PPP Multilink Protocol (MP) is supported under NT 4.0 and RAS and Windows 2000. PPP MP provides up to 128 kilobytes per second (kb/s) on a single BRI ISDN line by combining two B-channel calls. Each B-channel provides a 64 kb/s link. To attempt to use speeds greater than 64kb/s, you can modify the phone book entry by editing the entry, choosing the ISDN button, and entering 2 in the *Channels to Use* category.

5.4. An Example Using Data Transfer

To use the BRI/2 to transfer data, you can use any application that supports dial-up networking communication. For an example, you can set up two computers to share files through the Window Explorer.

5.4.1. PBX Setup

Requirements

- 2 computer with a BRI/2 installed and running
- ISDN PBX with 2 available extensions

NOTE: Windows NT Server is required on one of the host system if you want receive calls on multiple ports (PPP-MP). Refer to the *Installation and Configuration Guide* for more information.

Setup Procedure

Refer to *Figure 8* and perform the following:

1. Connect BRI/2 J1 of the Host Computer to the PBX (e.g., ext. 9156).
2. Connect BRI/2 J2 of the Host Computer to the PBX (e.g., ext. 9158).
3. Connect BRI/2 J1 of the Remote Computer to the PBX (e.g., ext. 9152).
4. Connect BRI/2 J2 of the Remote Computer to the PBX (e.g., ext. 9154).

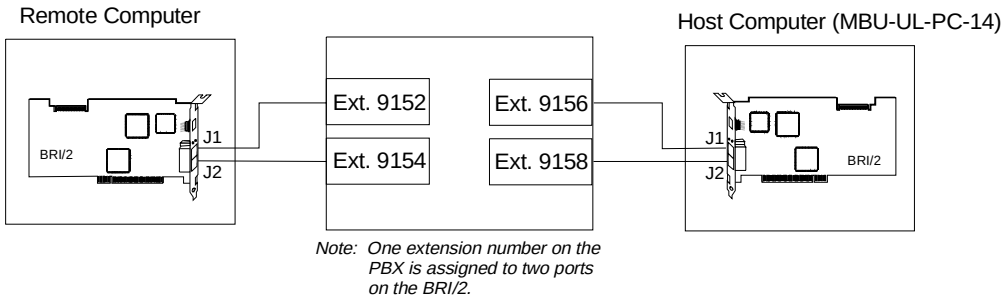


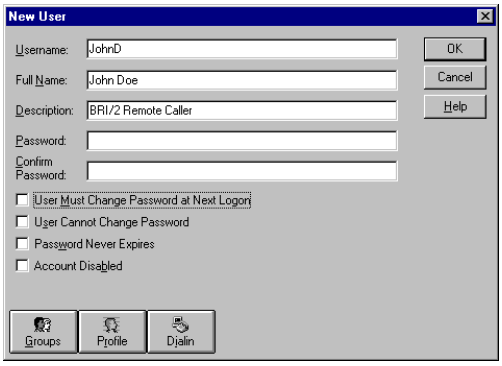
Figure 8. Data Transfer Test Setup

5.4.2. Computer Setup

On the Host Computer, perform the following:

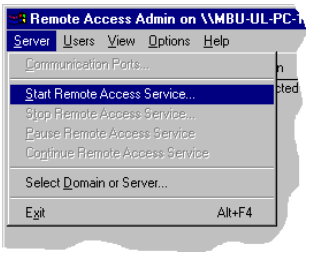
1. Run **User Manager** (click on Start/Programs/Administrative Tools).
2. Select **New User** under the *User* drop-down menu.

3. Add the name of the user (name and password) logged on to the Remote Computer.

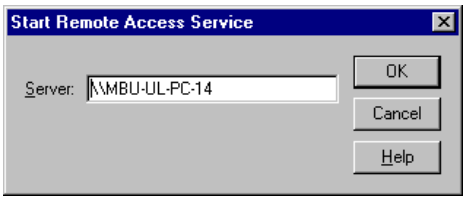


4. Run the **Remote Access Admin** (click on Start/Programs/Administrative Tools).

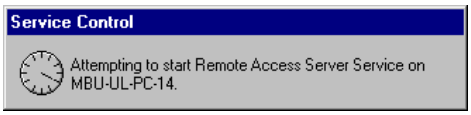
5. Select **Start Remote Access Service** under the *Server* drop-down menu.



6. The *Server Name* prompt is displayed and shows the name of the Host Computer (e.g., \\MBU-UL-PC-14). Click **OK**.

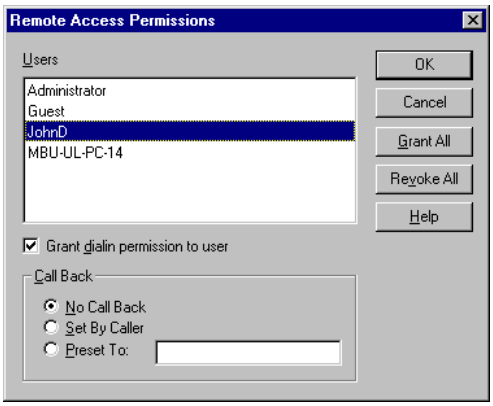


7. A progress indicator is displayed.



8. After the server is started, select **Permissions** under the *Users* drop-down menu.

9. Select **Grant dial in permission** for the user set up in Step 1.



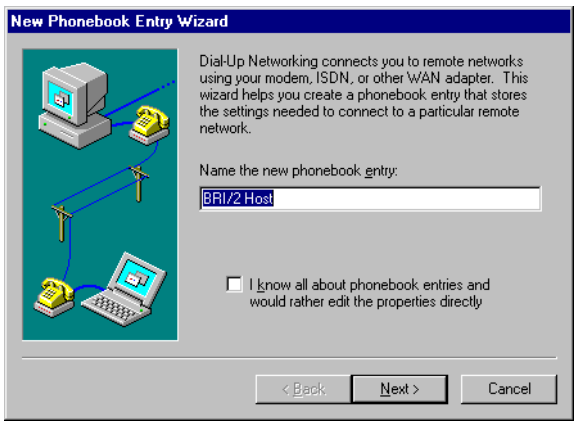
10. Click **OK** to close the *Remote Access Permissions* window.
11. Click **OK** to close the *Remote Access Admin* window.

On the Remote Computer, perform the following:

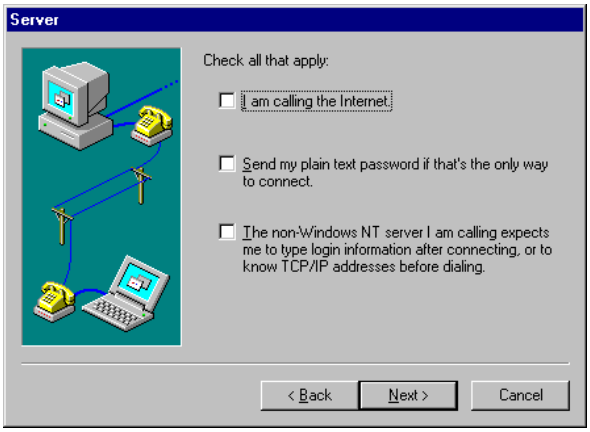
NOTE: If this is the first time you are using Dial-up Networking, you will skip Step 1.

1. In the *Dial-up Networking* applet, click the **New** button to add a new Phonebook entry.

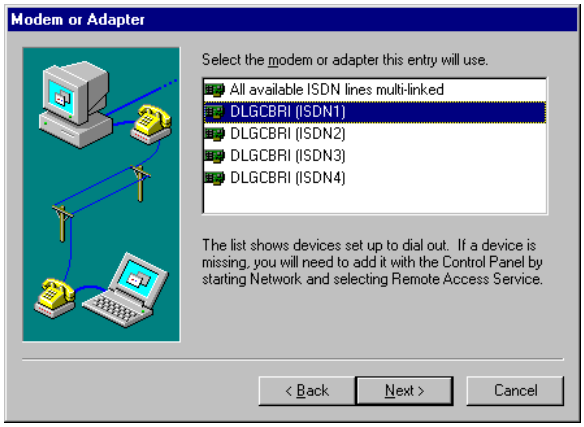
2. Type in a name for the connection and click **Next**.



3. In the *Server* window, do not check any options. Click **Next** to continue.

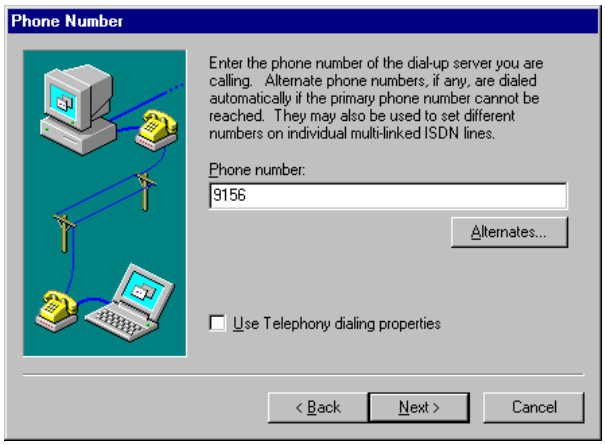


4. In the *Modem or Adapter* window, select one device [e.g., DLGCBRI (ISDN1)]. Click **Next**.



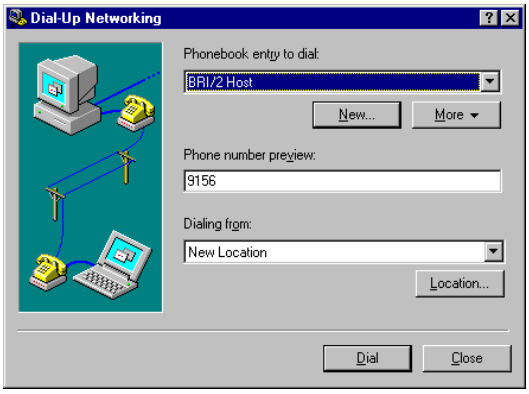
NOTE: In this example DLGCBRI (ISDN1) is used. You can select any valid BRI/2 device.

5. In the *Phone Number* window, type in the extension number connected to DLGCBRI (ISDN1) of Host Computer (refer to the Table below). Click **Next** to continue.

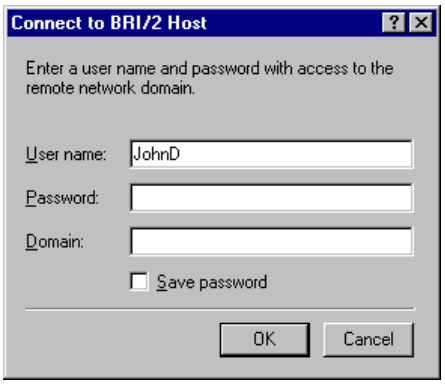


Host Computer		
BRI/2 Port	Device Name	PBX Extension
J1	DLGCBRI (ISDN1)	9156
J1	DLGCBRI (ISDN2)	9156
J2	DLGCBRI (ISDN3)	9158
J2	DLGCBRI (ISDN4)	9158

6. Click **Finish** and return to the *Dial-up Networking* window.



7. Click **Dial**.
8. When prompted, enter the user name and password and click **OK**.

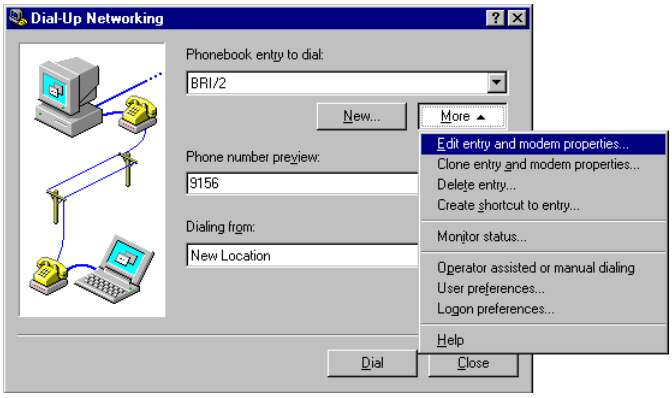


9. After the connection is established, you can view the Remote Computer in *Network Neighborhood*. You can now transfer files between the remote and host computers using Windows Explorer.

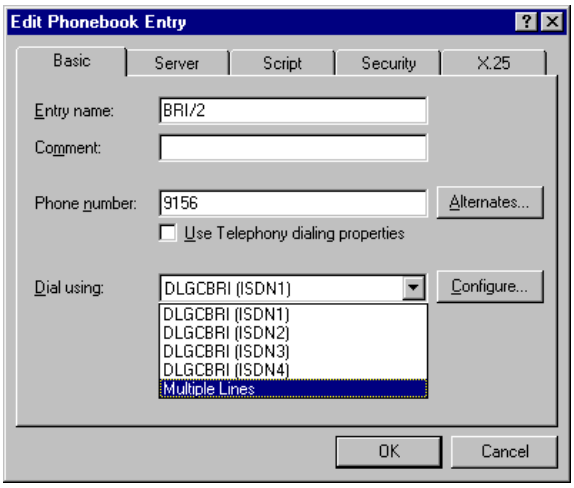
Multilink PPP

You can increase the connection speed by using multiple ports on the Remote Computer to connect to the Host Computer. On the Remote Computer, perform the following:

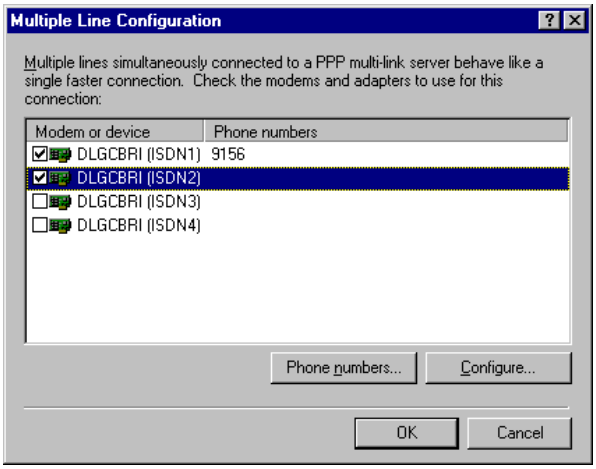
1. In the *Dial-up Networking* applet, for the Phonebook entry created in the previous example, click the **More** button, then select *Edit Entries and Modem Properties.*



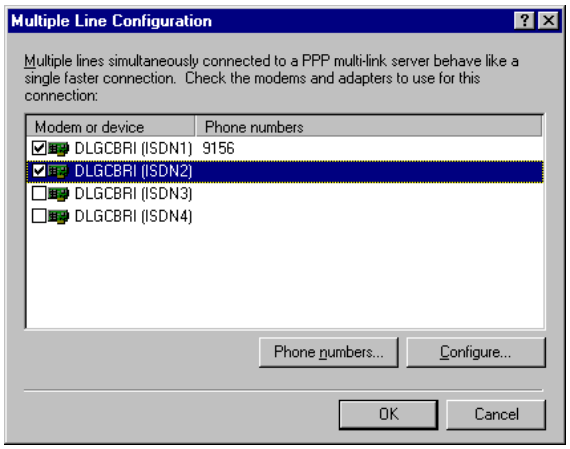
2. Under the *Basic* tab, next to Dial Using, select **Multiple Lines**. Then click **Configure**.



3. Select the device(s) you want to use.



4. For each device, click **Phone numbers** and type in the extension(s) of the Host Computer (refer to the Table below).



Remote Computer		
BRI/2 Port	Device Name	Extension to call
J1	DLGCBRI (ISDN1)	9156
J1	DLGCBRI (ISDN2)	9156
J2	DLGCBRI (ISDN3)	9158
J2	DLGCBRI (ISDN4)	9158

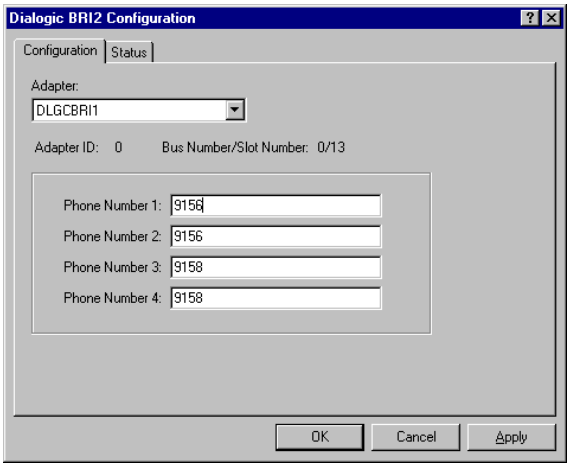
5. Click **Add**. Then click **OK** to close the *Phone Numbers* window.
6. Click **OK** to close the *Multiple Line Configuration* window
7. Click **OK** to close the *Edit Phonebook Entry* window. and return to the *Dial-up Networking* window.
8. Click **Dial**.
9. When prompted, enter the user name and password and click **OK**.
10. After the connection is established, you can view the Remote Computer in *Network Neighborhood*. You can now transfer files between the remote and host computers using Windows Explorer.

5.5. The Dialogic BRI/2 Configuration Utility

The *Dialogic BRI/2 Configuration* utility is copied to the Windows Startup directory when this package is installed. When you (re)start Windows, the *Dialogic BRI/2 Configuration* icon is displayed in the toolbar. Each of the four colored boxes represent a data port on the BRI/2. A red box (dark gray) indicates that a port is busy (a data connection is established). A green box (light gray) indicates that a port is idle (no connection). In the figure below, the icon shows that a data connection is established on port 1.

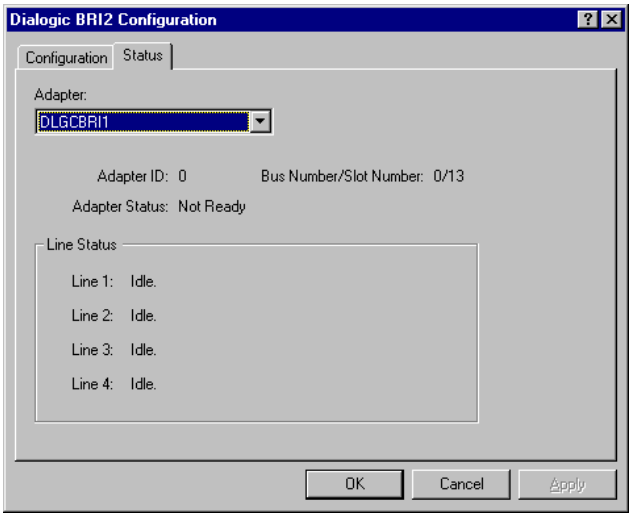
NOTE: The *Dialogic BRI/2 Configuration* icon displays status for data connections only. If there is a voice call on a port, the icon will show it as idle.

To display the *Dialogic BRI/2 Configuration properties*, double-click on the icon. The *Dialogic BRI/2 Configuration* window (shown below) is displayed.



The *Configuration* tab displays the phone numbers assigned to each port on a specific BRI/2 (listed in the *Adapter* field). Data entered in the *Phone Number* fields are used to send caller ID information to the host computer.

Click the *Setup* tab. The *Dialogic BRI/2 Configuration* window (shown below) is displayed.



The *Setup* tab displays the status of each port (line) for a specific BRI/2 (listed in the *Adapter* field). This status corresponds to the icon displayed in the toolbar.