{bmc novlogo.bmp} Modem Description Tool   Help

The following MDMWIN help topics are available.

Table of Contents

[1] WINDOWS

[2] MAIN_INDEX

[3] Help index

[4] Index;Help index

[5] Contents:000005

For Help on Help, Press F1

Disclaimer

Novell, Inc. makes no representation or warranties with respect to the contents or use of this manual, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose.   Further, Novell, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Novell, Incorporated
122 East 1700 South
Provo, Utah 84606

Trademarks

Novell has made every effort to supply trademark information about company names, products, and services mentioned in this document.   Trademarks indicated below were derived from various sources.

Hayes and Smartmodem are trademarks of Hayes Microcomputer Products, Inc.

NetWare and Novell are trademarks of Novell, Inc.

⁶ WINDOWS

⁷ Copyright

⁸ Copyright

Introduction

This specification defines how modem vendors can create modem description files that allow their modems to work with Novell's products.

Novell's most recent products and those in development are designed to be "modem independent". This allows new modems to be supported by these Novell products without a new version of the software being released.   All that the end user is required to do, is to load the appropriate modem descriptor file onto his system.

These Novell products can interpret modem description files,.   They execute scripts in the files to perform modem operations as the application requires.   Neither the modem control components nor the software products themselves are specific to any one modem or set of modems.   Any details specific to modems are contained in the modem description files.

This document describes the format and content of the information present in the modem description files.   The method of defining capabilities of a modem is specified and the process of constructing scripts to accomplish modem operations is outlined.   Multiple examples illustrate uses of the details presented in this specification.

Modem description files are created by the modem vendors using this specification.   Test programs will be provided by Novell Labs allowing vendors to check that the modem description functions correctly.   When the vendor is satisfied that the modem description file is correct, the file and modem can be certified by Novell Labs .   It will then be made available to our mutual customers as a supported modem.   It may also be integrated into future releases of Novell products.

When Novell's products are installed, modem description files are copied along with other product files.   As users configure the software they identify the modems to be used from lists of modem names.   Any modem that has a modem description will be presented in these lists for the user to select.

[9] WINDOWS

[10] Introduction

[11] Introduction

[12] Introduction

[13] Contents:000010

Description File Format

Modem descriptions are stored in files with the extension **.MDC**.   These files contain all the information, initialization strings, and responses for a set of modems.

---

[14] WINDOWS

[15] Description_File_Format

[16] Description File Format

[17] Description File Format

[18] Contents:000010

Overview

A modem description file includes information describing both a modem vendor and individual modems.   The information about the modem vendor is specified first with from one to many descriptions of modems following.

The vendor information begins with the vendor's name, which identifies the company creating the description file.   A copyright notice can be included to protect the company's rights.   Version information should be added to allow tracking of additions or corrections to description information.

Each file can describe more than one modem.   One way to organize modem descriptions is to collect information about all modems from one vendor into a single file.   This makes it easy to register the single filename with Novell. Another possibility is to group modems by 'family', as might be done with all of the "XYZ Ultra Xxxx" models. It is suggested that all modems manufactured by a vendor be located in a small number of files.

Modem description information begins with a line specifying the modem name.   This name must be unique within the entire set of modem names known to Novell, and thus should include some form of the vendor's name to avoid conflicting with any other vendor's descriptions.   The modem name will be used in lists presented to users and may be up to 39 characters in length.   Only one definition of a modem can exist across all modem description files; duplicates will produce unpredictable results.

The description for a modem continues with a line supplying information regarding the features, capabilities, and default values for the modem.   This information is needed by the modem control components to determine which logical operations may be performed.   Such information would include the highest interface bit rate possible to the modem, whether the modem can be used with leased lines, whether it supports use of hardware flow control, and so forth.   These details are supplied explicitly using the OPTIONS keyword.

The modem scripts that perform particular operations must be specified next.   These scripts are simply strings encoding sub-operations to be executed which together accomplish the desired operation.   Multiple lines can be combined if required.

The final section of a modem description will contain the strings used to decode a modem's responses to commands and operations.   For instance, the string returned by a modem when a command is successful must be associated with the meaning 'OK'.   Additional response recognition allows modem control components to handle call origination and answering.   These strings contain pairs of subsections, the first subsection giving the input to be recognized, and the second specifying the meaning of that input.

The modem description files that are delivered to the customer are in a binary format that is obtained by processing the ASCII text, as described in this document, by a conversion program to be supplied by Novell Labs to the modem vendor.   Novell Labs is also writing an interactive

[19] WINDOWS

[20] IDH_Overview

[21] Overview

[22] Overview;

[23] Contents:000035

program that will allow the editing and creation of these modem descriptor files.   The concepts described here apply equally well to both methods of creating the description file.   Chapter 4 describes these tools and their use.

Description Line Format

Specification lines are given in the form of a keyword followed by an equals sign and then one or more values.   For most of the keywords there will be only one string value. Other keywords, such as the OPTIONS keyword, can take multiple values which can be identifiers, numbers, and strings.

Control characters are included by prefixing with a reverse apostrophe, e.g. `M is used to enter a control-M.

ASCII 'nul' characters (\x00) must not be included in strings.   Also, script strings cannot contain the ASCII characters '[' and ']', except as noted below.

[24] WINDOWS

[25] Description_Line_Format

[26] Description Line Format

[27] Description Line Format

[28] Contents:000010

Vendor Description

The following keywords are valid in the vendor section:

- o   MANUFACTURER   Descriptive name of modem vendor
- o   COPYRIGHT   Vendor's copyright notice
- o   MAJOR_VER   Version number of modem description(s)
- o   MINOR_VER   Revision number of modem description(s)

The manufacturer and copyright string values may be up to 80 characters in length.   The version numbers may have numeric values from zero to 255.   Currently none of these values are directly used by modem control components, but are provided for use by modem vendors.   Examine the later examples to see typical uses of these keywords.

[29] WINDOWS

[30] Vendor_Description

[31] Vendor Description

[32] Vendor Description

[33] Contents:000010

Modem Description

This section descibes the different options which comprise a modem description.

---

34 WINDOWS

35 Modem_Description

36 Modem Description

37 Modem Description

38 Contents:000010

Modem Name

The modem name string value may be up to 39 characters in length.   As noted before, this name must be unique within the entire set of modem names known to Novell.

There can be multiple descriptions for the same modem with each appropriate for distinct circumstances.   For instance, it may be found that most revisions of a particular modem can be initialized quickly but that some ROM levels require delays between output characters.   Rather than force all users to wait for a lengthy initialization operation, it is possible to create two descriptions.

```
Modem Name = "XYZ Ultra Xxxx"
Modem Name = "XYZ Ultra Xxxx (Slow Init)"
```

[39] WINDOWS

[40] Modem_Name

[41] Modem Name

[42] Modem Name

[43] Contents:000010

Modem Options

Some options have a value associated with them, others indicate the presence or absence of a modem capability.   Options that have parameters are defined as shown here:

The following options require values to be defined:

o <u>DEFAULTRATE</u>   Best 'normal' bit rate to modem

o <u>DELAY</u>   Delay between command characters

o <u>FIXEDRATE</u>   Best bit rate for use with fixed rate usage (optional)

o <u>MAXRATE</u>   Maximum bit rate to modem

---

[44] WINDOWS

[45] Modem_Options

[46] Modem Options

[47] Modem Options

[48] Contents:000010

DEFAULTRATE

When a modem operation specifies use of fixed rate mode, the FIXEDRATE option supplies the bit rate to be used to communicate to the modem.   When that mode is not selected, modem control uses this option to determine the default bit rate for the interface to the modem.

[49] WINDOWS

[50] DefaultRate

[51] Default Rate

[52] Default Rate

[53] Contents:000010

DELAY

Some modems require a greater amount of time to process complex commands.   Experience has shown that complex commands that are output to these modems one character at a time will be successful.   This option supplies the amount of time to insert between characters on selected commands.

The numeric value is the time in tenths of seconds that modem control should wait between outputting characters.   There are two script operations for output: one inserting delays between characters with the other not delaying between characters.   If this option is not specified the default delay will be zero (and thus no delay at all).

[54] WINDOWS

[55] Delay

[56] Delay

[57] Delay

[58] Contents:000010

FIXEDRATE

Modems can be initialized to use one unchanging bit rate between themselves and the DTE.   This bit rate is usually set to a value high enough to permit use of compression no matter what line speed is used on a connection.   The numeric value is the bit rate to be used when the modem is put into fixed rate mode.

Note: This option also implies that fixed rates are supported by the modem (see FIXED script).

[59] WINDOWS

[60] FixedRate

[61] Fixed Rate

[62] Fixed Rate

[63] Contents:000010

MAXRATE

The set of interface bit rates that can be used to communicate from the DTE to a modem usually has an upper bound.   This option supplies the maximum interface bit rate to be used with a modem.   The numeric value for this option is the maximum rate in bits per second.

[64] WINDOWS

[65] MaxRate

[66] Max Rate

[67] Max Rate

[68] Contents:000010

Modem Scripts

A modem script is a text string that is sent to the modem to cause a particular activity.   They are associated with a particular modem capability and are transmitted to the modem when the application software wishes to invoke that operation.

More information on the content and creation of modem scripts will be given in a later section, "Script Operations".   Individual scripts are summarized here:

- o Err Ctrl Enable error control protocol(s)
- o Answer Place modem into auto-answer mode
- o Compression Enable data compression method(s)
- o Dial Originate a call on a switched line
- o Fixed DTE Place modem into fixed interface rate mode
- o Hangup Disconnect any call in progress
- o Flow Ctrl Place modem in hardware flow controlled mode
- o Initialize Initialize the modem to a known state
- o Leased Init Place the modem into leased line mode
- o Leased Ans Accept a leased line connection
- o Leased Org Originate a leased line connection
- o Manual Ans Accept manually answered switched connection
- o Manual Org Originate manually dialed switched connection

[69] WINDOWS

[70] Modem_Scripts

[71] Modem Scripts

[72] Modem Scripts

[73] Contents:000010

Error Control

This script enables the use of any of the error correcting protocols implemented by a modem when the next data connection is begun.   Since which protocols may be activated depends on the remote modem, this script only specifies that the best possible for each connection be used. Through monitoring of negotiation progress responses the modem control components can be informed of the characteristics of the protocol activated.

[74] WINDOWS

[75] Scr_ARQ

[76] Error Control Script

[77] ARQ;Error Control

[78] Contents:000010

Auto Answer

This script places the modem in the mode of automatically answering incoming telephone calls.   A connection can begin without intervention by modem control.   Modem control will monitor the progress of connection initiation and detect when the connection is complete and data transfer may begin.

[79] WINDOWS

[80] Scr_AutoAnswer

[81] Auto Answer Script

[82] Auto Answer

[83] Contents:000010

Compression

This script enables the use of any of the data compression methods implemented by the modem when the next data connection is begun.   Since the particular compression method employed depends partly on the remote modem, this script only specifies the preferred method to be used. Through monitoring of negotiation progress responses the modem control components can be informed of the characteristics of the method activated.

[84] WINDOWS

[85] Scr_Compression

[86] Compression Script

[87] Compression

[88] Contents:000010

Dialout

This script is executed when a call origination operation is requested on a switched line.   The operation request parameters include whether the dialing should use pulse or touch-tone signaling, and the destination telephone number.   These parameters are inserted into the dial script string using the substitution tags, "[T]" and "[P]".   These tags are described more fully in a later section.

[89] WINDOWS

[90] Scr_Dialout

[91] Dialout Script

[92] Dialout

[93] Contents:000010

Fixed DTE

This script places a modem into fixed interface bit rate mode.   This allows the interface to be programmed to one bit rate that can be used for all subsequent connections.   The actual rate used will be determined by the associated FIXEDRATEoption.

[94] WINDOWS

[95] Scr_Fixed

[96] Fixed Rate Script

[97] Fixed Rate

[98] Contents:000010

Hangup

This script causes the modem to disconnect any call that might be in progress. (i.e.. place the modem "on hook")   This script should specify all needed operations that to ensure the call is disconnected, irrespective of the current modem state.

[99] WINDOWS

[100] Scr_Hangup

[101] Hangup Script

[102] Hangup

[103] Contents:000010

Hardware Flow Control (RTS/CTS)

This script places a modem into hardware flow controlled mode.   In this mode data transfer between modem and interface is controlled through the use of the RS-232 signals Request-to-Send (RTS) and Clear-to-Send (CTS).   Each signal controls data transfer in one direction.

[104] WINDOWS

[105] Scr_HWFC

[106] Hardware Flow Control Script

[107] Hardware Flow Control

[108] Contents:000010

Initialize

This script causes a modem to be initialized to a known state.   This state must have all optional features disabled.   That is, the purpose of the INIT script is to put the modem into a state where any of the other features can then be added by individually executing scripts.

The INIT script is usually the first script executed when a modem operation is begun; the only script that could precede it might be the HANGUP script to disconnect a call in progress.   The INIT script can make no assumptions about the previous state of the modem.   Indeed, the previous user of a modem may not have been Novell's modem control, and so not even modem control will know the state of a modem.

The script must reset everything that can be affected by modem commands.   This would include features like echo, call progress, result code, modem signal usage, flow control modes, and so forth. The script must set the correct modes so that modem response strings can be recognized.

[109] WINDOWS

[110] Scr_Init

[111] Initialize Script

[112] Initialize Script

[113] Contents:000010

Leased Line Init

When a modem initialization operation is requested and the leased line feature is requested, this script is executed to place the modem into leased line mode.   In some cases this feature is not under control by commands, but rather some switches must be set.   In this case the script may be absent.

[114] WINDOWS

[115] Scr_LsdInit

[116] Leased Line Initialize Script

[117] Leased Initialize

[118] Contents:000010

Leased Line Answer

This script is executed when a call answer operation is requested on a leased line.   The modem should attempt to connect to the remote modem using answering frequencies.   Once this script is completed modem control will monitor the local modem's responses to detect when a connection has begun.

[119] WINDOWS

[120] Scr_LeasedAnswer

[121] Leased Line Answer Script

[122] Leased Answer

[123] Contents:000010

Leased Line Originate

This script is executed when call origination is requested on a leased line.   The modem should attempt to connect to the remote modem using origination frequencies.   Once this script is completed modem control will monitor the local modem's responses to detect when a connection has begun.

[124] WINDOWS

[125] Scr_LeasedOriginate

[126] Leased Line Originate Script

[127] Leased Originate

[128] Contents:000010

Manual Answer

This script is executed when a manual call answer operation is requested.   The modem should attempt to connect to the remote modem using answering frequencies.   Once this script is completed modem control will monitor the local modem's responses to detect when a connection has begun.

---

[129] WINDOWS

[130] Scr_ManualAnswer

[131] Manual Answer Script

[132] Manual Answer

[133] Contents:000010

Manual Originate

This script is executed when a manual call origination is requested.   The modem should attempt to connect to the remote modem using origination frequencies.   Once this script is completed modem control will monitor the local modem's responses to detect when a connection has begun.

---

[134] WINDOWS

[135] Scr_ManualOriginate

[136] Manual Originate Script

[137] Manual Originate

[138] Contents:000010

Modem Responses

The response strings in a modem description allow recognition and interpretation of data sent from the modem to the DTE.   This informs the modem control software of the success or failure of a command.   It also lets modem control detect when a call is arriving.

As the responses generated differ between modems, the modem vendor must supply information to allow modem control to recognize responses.   Response strings contain from one to many pairs of sub strings, the first giving the input string to be recognized and the second representing the standard 'meaning' of the string

With the ever more complex responses found in newer modems, it is sometimes necessary to perform multi-stage matching of response strings.   As an example, when using negotiation progress monitoring to capture added information about connections, the PROTOCOL response can be received.   The first stage of recognition would identify the input as the PROTOCOL message. The second stage of recognition would then identify the particular sub strings that may be present in this message.

Modem control will accumulate ASCII characters received from a modem until a carriage return character ('\x0D' or decimal 13) is received; all other control characters are ignored.   The accumulated string is then compared to the match strings in the RESPONSES keyword string. When a match is found the meaning is interpreted and the appropriate action is taken.

The following response scripts are defined:

- o   Responses
- o   Substring 1
- o   Substring 2
- o   Substring 3
- o   Substring 4

The first script provides the first stage matching information.   Part of the meanings information can specify that input matching for a message is to continue using information found in one of the other four response string keywords.

Further information on response matching is given in the section "Response Strings" below.

[139] WINDOWS

[140] Responses

[141] Responses

[142] Responses

[143] Contents:000010

Script Operations

A modem script contains a sequence of nano-operations that tell modem control which actions to perform.   These actions include output of ASCII characters, controlling interface signals, checking for expected input, and so forth.   There is no facility   for conditional execution of nano-operations; the entire script is executed unless an error occurs.

Each nano-operation consists of an alphabetic character optionally followed by parameters for that operation.   These values may be string and/or time values, or other modifiers for that basic operation.   The operations are summarized below:

- o   Toggle Break
- o   Toggle DTR
- o   Flush Buffers
- o   Wait for Input (must match)
- o   Conditional Input (optional match)
- o   Output string
- o   Output w/ delay (delay between characters)
- o   Quiet Wait (idle time on DTE)
- o   Pause
- o   Change Data Rate   (change DTE interface rate)

[144] WINDOWS

[145] Operations

[146] Script Operations

[147] Script;Operations

[148] Contents:000010

Toggle Break

This operation is used to turn on the asynchronous 'break' signal momentarily.   This might be used to get a modem's attention to switch it into command mode.

The break operation character may be followed by a decimal number giving the length of time in tenths of seconds for which break is to be turned on.   If a time value is not given then the default break on time of one half second will be used.

---

[149] WINDOWS

[150] Toggle_Break

[151] Toggle Break

[152] Break

[153] Contents:000010

Toggle DTR

This operation is used to control the DTR signal to the modem.   The DTR signal will be turned off momentarily and then turned on again.   Turning off this signal might be used to get the attention of a modem when it is in data transfer mode.

If a time value is not given then the default DTR off time of one half second will be used.

---

[154] WINDOWS

[155] Toggle_DTR

[156] Toggle DTR

[157] DTR

[158] Contents:000010

Set Interface Data Rate

This operation allows scripts to change the data rate used to communicate with the modem.   This is used with modems that do not automatically resynchronize interface data rates after switching back to command mode from data transfer mode.

After execution of this operation any further output or input through the interface will use this data rate.   Some asynchronous equipment must flush one or both of the input and output streams when changing data rates.

[159] WINDOWS

[160] ChangeRate

[161] Change DTE Rate

[162] Change DTE Rate

[163] Contents:000010

Flush Buffers

Characters that have been buffered for output or input but not yet processed may be discarded by this operation.   This might be useful when modem responses up to a point may safely be ignored, or if prior output should be discarded when starting new commands.

The flush operation can be set to flush any combination of input and output buffers.

[164] WINDOWS

[165] Flush_Buffers

[166] Flush Buffers

[167] Flush Buffers

[168] Contents:000010

Input String

This operation allows a script to check for a specific string to be received from a modem.   For example, after most modem commands a script should check for the returned indication of success, usually 'OK'.   There are two variants : 'must match' or 'conditional match'

The operation can optionally be followed by a decimal number specifying the maximum time to wait for this   response.   This value is specified in tenths of a second.   If it is not given the default value of five seconds is used.

Also needed is a delimiter character.   This is the character which sets the beginning and end of the input   string.   The delimiter is a character which is not used in the actual text of the input string.

Modem control continues receiving characters from the modem until one of two occurrences.   If a matching string from the modem is completed the nano-operation finishes and the script continues. If a match is not completed and the time-out period has elapsed since the last character was received from the modem, then an input time-out is declared.

If this was a 'must match' input string operation the time-out causes the script to be terminated with a 'bad modem response' error code.   Otherwise the time-out simply terminates the optional match operation and continues with the rest of the script.

---

[169] WINDOWS

[170] Input_String

[171] Input String

[172] Input

[173] Contents:000010

Output String

This operation allows output of character strings from the script to a modem.   The output string may contain any non-nul, non-control ASCII characters. There are two types of output operations, a 'straight output' and 'output w/ delay'. If it is necessary to insert a delay between characters the 'output w/ delay' operation will use the delay time specified using the DELAY option.

The string to be output is bounded by some character chosen by the script creator. The character directly following the operation code will be used to locate the end of the string.   The script creator should choose a string delimiter character which will not be used for any interactions with the modem. The character should not be one of the alphanumerics as this would make reading descriptions difficult.   A survey of several modems has identified the many punctuation characters that are used within modem commands and responses.   This leaves the following set as the characters recommended for use: '<^_{}|:', by convention a colon is used (":").

As noted in section "Description Line Format", control characters may be inserted into output strings   using the accent-grave or reverse apostrophe.

Variable strings may be substituted into output or input strings by use of an substitution marker.   A substitution is indicated by a substitution tag name surrounded by [and] characters.   For example, substitution of the "tone" or "pulse" modifier and the phone number into a dial-out command might be coded as "ATD[T][N]'M" where [T] is replaced with 'T' or 'P' and [N] is replaced with a dial number.

Only a limited number of substitution tags are defined, and the substituted strings are not variable by modem type. The pre-defined tags are:

- o   T   dial tone/pulse modifiers: 'T' or 'P'
- o   N   dial phone number: supplied by application
- o   R   ring count: used on initialization
- o   W   ring count: used on initialization

Care should be taken that the longest command sent to a modem does not exceed what the modem can handle.   Many modems are limited to a maximum of 40 command characters excluding the leading 'AT', spaces, hyphens, and final carriage return.   The input command can be used to break up long command output sequences.

---

[174] WINDOWS

[175] Output_String

[176] Output String

[177] Output

[178] Contents:000010

{bmc novlogo.bmp} Pause

Pause

This operation allows a script to pause execution for a period of time.   This is useful when modems may require some time to complete complicated modem commands. A time value can be given which determnes how long (in tenths of a second) the pause will last.   If a time value is not given then the default time value of one second is used (time = 10).

[179] WINDOWS

[180] Pause

[181] Pause

[182] Pause

[183] Contents:000010

Quiet

This is used to skip all the responses from a previous command before issuing a new command.   It causes a wait until the modem remains continuously quiet for the specified time.

A time value can be given which determnes how long (in tenths of a second) the pause will last.   If a time value is not given then the default time value of one second is used (time = 10).

This operation will throw away any data received from the modem.   Whenever a character is received the elapsed time timer will be reset to zero.   When the elapsed time timer reaches the specified wait time value the nano-operation completes successfully.   An additional timer records the total time since the nano-operation began.   If this timer reaches the sum of the specified wait time plus five seconds then a time-out is declared and the nano-operation completes unsuccessfully causing the script to be terminated with an error.

[184] WINDOWS

[185] Quiet_Wait

[186] Quiet_Wait

[187] Quiet_Wait

[188] Contents:000010

Response Strings

[189] WINDOWS

[190] Response_Strings

[191] Response Strings

[192] Response Strings

[193] Contents:000010

Match Strings

The first of each pair of strings in the RESPONSES string is known as the match string. As noted above, when modem control is monitoring modem responses, characters received from a modem are collected until a carriage return is received. The input string is then compared (case insensitive) against all the match strings found in RESPONSES keyword strings. This matching operation proceeds in the same order as the RESPONSES string occurred in the description file.

Match strings do not need to be the entire response string to declare a match. Only the initial characters of a response must match the match string. Thus the match string "ERR" will match both the response strings "ERROR" and "ERRONEOUS" but not "ERASE". However, this may make the order that match strings are tried even more important.

[194] WINDOWS

[195] Match_Strings

[196] Match Strings

[197] Match Strings

[198] Contents:000010

Response Meanings

The second string of each pair of strings is known as the meanings string.   The interpretation of this string defines what the recognized response means to modem control.   This includes whether the response is a success, failure, or some intermediate indication.   When certain optional connection features are recognized they can be signaled to modem control by this method.   Finally, this is the way that bit rates are given to modem control.

There are four types of meanings information: status, feature, rate, and match chaining.   The status and feature values are decimal indices into tables used by modem control.   The rate decimal value is the actual data rate in bits per second.   The match chaining value will be described later.

Status   Report Status; may terminate scanning.

Rate   Bit rate of connection.

Features Reports an enabled feature for this connection.

Chaining Continues scanning of the response in the selected substring.

[199] WINDOWS

[200] Response_Meanings

[201] Response Meanings

[202] Response Meanings

[203] Contents:000010

Status Meanings

Status information is used to notify modem control when something of significance has been discovered in a response, or to report that scanning should continue.   The following status index values are defined:

.	Status Values		.
none	IGNORE	INFO1	INFO2
INFO3	INFO4	OK	RING
RRING	CONNECT	BUSY	NO ANSWER
NO CARRIER	ERROR	NO DIALTONE	VOICE
UNKNOWN			

---

[204] WINDOWS

[205] Status_Meanings

[206] Status Meanings

[207] Status Meanings

[208] Contents:000010

Rate Meanings

The rate meaning is used to tell modem control what the current line data rate is in bits per second. For most modems that implement negotiation progress messages this rate value can be captured from the "CARRIER <R>" response by using the **<R>-value** selection.   This will match any speed response from the modem, and capture that value to return it in the rate definition command.

.   Rate Values   .

50	75	110	134.5
150	300	600	1200
1800	2000	2400	3600
4800	7200	9600	19200
38400	57600	115200	<R>-Value

[209] WINDOWS

[210] Rate_Meanings

[211] Rate Meanings

[212] Rate Meanings

[213] Contents:000010

Features Meanings

The feature values indicate to modem control when optional connection features have been enabled on the current connection.   Information on which features are enabled or disabled is made available to applications.   Applications can use this to determine whether they must independently perform error control or data compression for a connection.   The features values are summarized below:

Features

error control

MNP 5

V.42bis

unbalanced

synchronous

---

[214] WINDOWS

[215] Feature_Meanings

[216] Feature Meanings

[217] Feature Meanings

[218] Contents:000010

Match Chaining

The match chaining directs modem control to continue matching using the remainder of the input string (after the initially matched portion) and using a different substring.   This permits the multi-stage matching that is so useful with complex sets of responses like negotiation progress messages.

For the example response string : "PROTOCOL: ERROR-CONTROL"

The first string is part of the first stage matching string formed from the Responses list.   Modem control interprets it to mean that the response beginning with "PROTOCOL" is not a final response, rather that additional matching must be done using the **Subtring1** list.

Modem control begins repeatedly checking the remainder of the input string against the **Substring1** match strings.   Each time the match strings are used up modem control advances to the next character in the input string and tries again.   This continues until all the characters in the input string have been exhausted.   In this manner modem control finds the "ERROR-CONTROL" substring and notes that feature one, **error control**, is enabled for this connection.

---

[219] WINDOWS

[220] Match_Chaining

[221] Match Chaining

[222] Match Chaining

[223] Contents:000010

Environments

Novell's modem control is being implemented in multiple environments.   A short description of how modem description files are used in each environment is given here.

[224] WINDOWS

[225] Environments

[226] Environments

[227] Environments

[228] Contents:000010

NetWare Server

Modem description files on a Novell NetWare server are placed in a subdirectory accessible to NLMs (NetWare Loadable Modules).   At present there are two such products on NetWare servers: NetWare Connect and NetWare MultiProtocol Router.

NetWare Connect

The modem control components of NetWare Connect exist in an NLM called SYS:SYSTEM\AIO.   All files containing compiled modem descriptions are copied to this subdirectory.   When AIO is loaded it searches this subdirectory for files with the extension MDC (*.MDC).   AIO then creates a list of all modem names defined in these files and which file contains the description for each.   When one of NetWare Connect's services attempts a modem operation on a port, AIO determines which modem is attached to that port and ensures that modem's description has been read into memory.   AIO then starts the execution of the operation using the service's request parameters and the modem description.

NetWare MultiProtocol Router

The MultiProtocol Router uses uncompiled modem definition files that are placed in the SYSTEM directory.   It interprets these files as needed for modem control

[229] WINDOWS

[230] Env_Server

[231] NetWare Server Environment

[232] Server Environment

[233] Contents:000010

DOS

Modem description files in the DOS environment are usually installed in the same directory with the DOS program that uses them.   One such program is DIALCON, the connection establishment utility for the NetWare Remote Node (NRN) product in NetWare Connect.

DIALCON is executed by users when they wish to connect or disconnect the telephone line. DIALCON is also used to configure the parameters for using asynchronous connections; these parameters include the currently selected modem.   This information is saved in the NET.CFG file also used by other client network programs.

When a user starts DIALCON, it looks up the current modem name and then locates the file containing that modem description.   The description is loaded into memory and the modem control components perform modem operations using the description.   After the requested operations are complete, DIALCON terminates and the in-memory modem description is discarded.

---

[234] WINDOWS

[235] Env_DOS

[236] DOS Environment

[237] DOS Environment

[238] Contents:000010

Development & Testing Tools

Novell Labs is in the process of creating development and testing tools for vendors implementing modem descriptor files.   They will be made available to developers as they are completed, together with example modem descriptor files.

Novell Labs will also be offering a certification service for modems; the details of which are still being decided.

Please contact Novell Labs at (408) 577-7201 for up-to-date information.

[239] WINDOWS

[240] Development_Tools

[241] Development & Testing Tools

[242] Development;Tool;Testing

[243] Contents:000010