

CdBase Control

The CdBase control handles all the hard stuff associated with the Internet CDDb database. It also can vastly simplify the control of a CD player. It can run invisible to the user allowing the application to have it's own unique look and feel.

Properties

[AppName](#)
[Artist](#)
[Categories](#)
[Category](#)
[Cdid](#)
[CdTitle](#)
[CloseMatchWinLeft](#)
[CloseMatchWinTop](#)
[Command](#)
[ConnectTimeout](#)
[CurrentTrack](#)
[DataRev](#)
[Email](#)
[ExtData](#)
[HostName](#)
[Mode](#)
[PlayOrder](#)
[Port](#)
[Revision](#)
[Server](#)
[Sites](#)
[Title](#)
[Tracks](#)
[UseFirstMatch](#)
[Visible](#)

Methods

[GetCategory](#)
[GetCddb](#)
[GetCddbUseQueryString](#)
[GetCdDriveLetters](#)
[GetIsCat](#)
[GetMotd](#)
[GetMSid](#)
[GetMyHostName](#)
[GetQueryString](#)
[GetSites](#)
[NewLineFix](#)
[PlayTrack](#)
[Position](#)
[SetExtTrackData](#)
[SetSongName](#)
[SiteAddress](#)
[SiteLocation](#)
[SitePort](#)
[SongLength](#)
[SongName](#)
[SubmitString](#)

Events

[DataSent](#)
[ModeChange](#)

AppName Property

Returns or sets the name of the application.

Syntax

Object.**AppName** = string

The **AppName** property syntax has these parts:

Part	Description
<i>Object</i>	The CdBase object.
<i>String</i>	The name of the application. Default = VbActiveX It cannot contain any blank spaces.

Data Type

String

Remarks

This is sent to the cddb server. It should be set to the name of your application. This will allow the cddb server to distinguish different programs using the same control.

Example

```
Private Sub Form_Load()  
    CdBase1.AppName = "MyCd"  
    CdBase1.Email = "me@mine.com"  
End Sub
```

Artist Property

Returns or sets the name of the CD artist.

Syntax

Object.**Artist** = *String*

The **Artist** property syntax has these parts:

Part	Description
<i>Object</i>	The CdBase object.
<i>String</i>	The name of the artist

Data Type

String

Remarks

The control attempts to pull the artist name from the title if it cannot, it will return the full title

Example

Categories Property

Returns the quantity of categories found from a lscat (list categories) command

Syntax

Object.Categories = *Integer*

The **Categories** property syntax has these parts:

Part Description

Object The CdBase object.

Integer The number of categories available.
 Default = 0

Data Type

Integer

Remarks

The value is 0 until a successful Getlscat (get categories) method has occurred. After that, the value is the number of categories available.

Example

This creates a string that is a list all of the categories and shows it in a message box. This assumes the category list has already been retrieve via Getlscat.

```
Private Sub Command1_Click()  
    Dim I As Integer  
    Dim NumCategories As Integer  
    Dim CatList as String  
    NumCategories = CdBase1.Categories 'gets number of categories  
    For I = 1 To NumCategories  
        CatList = CatList + CdBase1.GetCategory(I) + vbCrLf  
    Next I  
    MsgBox CatList  
End Sub
```

Category Property

Returns or sets the category of the current CD.

Syntax

Object.**Category** = *String*

The **Category** property syntax has these parts:

Part Description

Object The CdBase object.

String The name or address of the cddb server.
Default = (empty)

Data Type

String

Remarks

This starts as an empty string. After a successful read from the cddb server, the value is set to what was read from the server.

Cdid Property

Returns (read only) the calculated CDID for the CD in the drive

Syntax

Object.Cdid = String

The **Cdid** property has these parts

Part	Description
<i>Object</i>	The CdBase object
<i>String</i>	The calculated CDID

Data Type

String

Remarks

The **Cdid** is calculated each time you read this property. It may take a few seconds to do depending on the state of the CD. Statements like `HalfCdid = Left(Cdid,len(Cdid) / 2)` will actual read and calculate the **Cdid** twice and take a long time. It would be better to read it into a temporary variable first in the above case.

Example

```
Private Sub Command1_Click()  
    Dim ThisCdid As string  
    ThisCdid = CdBase1.Cdid  
    MsgBox ThisCdid  
End sub
```

CdTitle Property

Returns or sets the name of the title of the CD.

Syntax

Object.CdTitle = *String*

The CdTitle property syntax has these parts:

Part	Description
<i>Object</i>	The CdBase object.
<i>String</i>	The title of the CD.

Data Type

String

Remarks

The control attempts to pull the Cd title from the title if it cannot, it will return the full title.

CloseMatchWinLeft Property

Returns or sets the distance between the left edge of close match selection window and the left edge of the screen.

Syntax

Object.**CloseMatchWinLeft** = *Long*

The CloseMatchWinLeft property syntax has these parts:

Part	Description
<i>Object</i>	The Cdbase object.
Long	The location of the left edge Default 0 (see remarks)

Data Type

Long

Remarks

This value is available if you want to control the location of the close match selection window. If the default value of zero is used it will pop up in the center of the screen.

CloseMatchWinTop Property

Returns or sets the distance between the top edge of close match selection window and the top edge of the screen.

Syntax

Object.**CloseMatchWinTop** = *Long*

The CloseMatchWinTop property syntax has these parts:

Part	Description
<i>Object</i>	The CdBase object.
Long	The location of the top edge. Default 0 (see remarks)

Data Type

Long

Remarks

This value is available if you want to control the location of the close match selection window. If the default value of zero is used it will pop up in the center of the screen.

Command Property

Specifies a CD player command to execute. This property is write only.

Syntax

Object.Command = *String*

The **Command** property has these parts

Part	Description
<i>Object</i>	The CdBase object
<i>String</i>	The command to be executed

Remarks

The string gives the name of the CD player command to execute: Open, Close, Play, Pause, Stop, Prev, Next, or Eject. The command is executed immediately. The following table describes each command. All commands other than “Open” assume the device is open.

Command	Description
<i>Open</i>	Opens the CD player device.
<i>Close</i>	Closes the CD player device.
<i>Play</i>	Plays the CD at the current location
<i>Pause</i>	Pauses CD player
<i>Stop</i>	Stop the CD player
<i>Prev</i>	Goes to the beginning of the current. If executed within three seconds of the previous prev command, goes to the beginning of the previous track or to the beginning of the first track if at the first track.
<i>Next</i>	Goes to the beginning of the next track (if at last track, goes to beginning of last track) using the Seek command.
<i>Eject</i>	Ejects the CD.

Data Type

String

Example

```
Private Sub Form_Load()  
    CdBase1.Command = "Open" 'Open the CD device  
End Sub  
  
Private Sub Command1_Click()  
    CdBase1.Command = "Play" 'Start playing the CD  
End Sub
```

ConnectTimeout Property

Reads or sets the connection timeout value.

Syntax

Object.ConnectTimeout = Long

The **ConnectTimeout** property has these parts

Part	Description
<i>Object</i>	The Cdbase object
Long	The connection timeout in milliseconds Default = 30000 Range 5000 - 65,535

Data Type

long

Remarks

The ConnectTimeout property is the time the program will wait for a response from a cddb server after sending a command or trying to connect.

Example

```
Private Sub Command1_Click()  
    Cdbase1.ConnectTimeout = Val(Text1.Text)  
End Sub
```

CurrentTrack Property

Reads or sets the current track number.

Syntax

Object.**CurrentTrack** = *Integer*

The **CurrentTrack** property has these parts

Part	Description
<i>Object</i>	The CdBase object
<i>Integer</i>	The current CD track number.

Data Type

Integer

Remarks

If the CD player is currently playing, the CD will stop and move to the CurrentTrack number. When the player is started again, it will start at the specified track. If the CD device is not open the setting this property will do nothing. If the track number is not a valid number, nothing will happen.

Example

```
Private Sub Command1_Click()  
    CdBase1.CurrentTrack = Val(InputBox("Track number", "MyApp", "1"))  
End Sub
```

DataRev Property

Reads or sets the revision number of the cddb database entry.

Syntax

Object.DataRev = Integer

The **DataRev** property has these parts

Part	Description
<i>Object</i>	The CdBase object
<i>Integer</i>	The revision number.

Data Type

Integer

Remarks

The cddb server allows updates to it's database. The revision number represents the revision of the data. The should be increased if you wish to submit a revision.

Email Property

Returns or sets the **email** address of the user.

Syntax

Object.Email = Integer

The **Email** property syntax has these parts:

Part	Description
<i>Object</i>	The CdBase object.
<i>String</i>	the email address of the user. Default = empty string

Data Type

String

Remarks

This is sent to the cddb server. The property must contain a value before the cddb server can be accessed. The cddb people request that this contain the users real **email** address and that programmers not hardwire a default value into their programs.

Example

```
Private Sub Form_Load()  
    CdBase1.AppName = "MyCd"  
    CdBase1.Revision = "1.0"  
    CdBase1.Email = "me@mine.com"  
End Sub
```


HostName Property

Returns or sets the HostName of the client computer

Syntax

Object.**HostName** = *String*

The **HostName** property syntax has these parts:

Part	Description
<i>Object</i>	The Cdbase object.
<i>String</i>	The hostname

Data Type

String

Remarks

This must be set before the control can handshake with the server. Host name of client. Example: abc.fubar.com

ExtData Property

Returns or sets the extended data for the CD

Syntax

Object.**ExtData** = *String*

The **ExtData** property syntax has these parts:

Part	Description
<i>Object</i>	The CdBase object.
<i>String</i>	The extended data.

Data Type

String

Remarks

This field contains the "extended data" for the CD. This is intended to be used as a place for interesting information related to the CD, such as credits, et cetera.

Mode Property

Returns the **mode** of the CD player

Syntax

Object.**Mode** = string

The **Mode** property syntax has these parts:

Part	Description
<i>Object</i>	The CdBase object.
<i>String</i>	The Cd Player Mode

Data Type

String

Remarks

Value	Description
524	Device is not open.
525	Device is stopped.
526	Device is playing.
529	Device is paused.
530	Device is ready.

PlayOrder Property

Returns or sets the **PlayOrder** string.

Syntax

Object.PlayOrder = *String*

The **PlayOrder** property syntax has these parts:

Part	Description
<i>Object</i>	The CdBase object.
<i>String</i>	The play order

Data Type

String

Remarks

This field contains a comma-separated list of track numbers which represent a programmed track play order. This field is generally stripped of data in non-local database entries. Applications that submit entries for addition to the main database should strip this keyword of data.

Port Property

Returns or sets the port of the server.

Syntax

Object.**Server** = *Integer*

The **Integer** property syntax has these parts:

Part	Description
<i>Object</i>	The CdBase object.
<i>String</i>	The name or address of the cddb server. Default = 8880

Data Type

String

Remarks

The default is the port of the “always there” server at cddb.cddb.com

Example

```
Private Sub cmdAccept_Click()  
    CdBase1.Server = txtAddress.Text  
    CdBase1.Port = Val(txtPort.Text)  
End Sub
```

Revision Property

Returns or sets the revision of the application.

Syntax

Object.**Revision** = *String*

The **Revision** property syntax has these parts:

Part	Description
<i>Object</i>	The CdBase object.
<i>String</i>	The revision of the application. Default = 1.0

Data Type

String

Remarks

This is sent to the cddb server in the hello command. It should be set to the **revision** of your application. It must not contain any blanks

```
Private Sub Form_Load()  
    CdBase1.AppName = "MyCd"  
    CdBase1.Revision = "1.0"  
    CdBase1.Email = "me@mine.com"  
End Sub
```

Server Property

Returns or sets the address of the server. It can be in the host name format such as “cddb.cddb.com” (preferred) or the IP address string format such as “100.0.0.1”.

Syntax

Object.**Server** = *String*

The **Server** property syntax has these parts:

Part	Description
<i>Object</i>	The CdBase object.
<i>String</i>	The name or address of the cddb server. Default = cddb.cddb.com

Data Type

String

```
Private Sub cmdAccept_Click()  
    CdBase1.Server = txtAddress.Text  
    CdBase1.Port = Val(txtPort.Text)  
End Sub
```

Sites Property

Returns (read only) an integer representing the number of sites received in the last sites query.

Syntax

Integer = *Object*.Sites

The **Sites** property syntax has the following parts

Part	Description
<i>Object</i>	The CdBase object
<i>Integer</i>	The number of sites

Data Type

Integer

Remarks

This number is set after the GetSites method has be called

Example

```
Private Sub Command1_Click()  
    Dim I As Integer  
    Dim NumSites As Integer  
    Dim SList as String  
    NumSites = CdBase1.GetSites  
    NumSites = CdBase1.Sites 'also returns sites  
    For I = 1 To NumSites  
        SList = SList + CdBase1.SiteLocation(I) + vbCrLf  
    Next I  
    MsgBox SiteList  
End Sub
```

Title Property

Returns or sets the string that represents the CD title.

Syntax

Object.**Title** = *String*

The **Title** property syntax has the following parts

Part	Description
<i>Object</i>	The CdBase object
<i>String</i>	The title of the CD

Remarks

The title is usually, but not always in the form Artist / CdTitle.

Tracks Property

Returns (read only) an integer representing the number of tracks on the CD.

Syntax

Integer = CdBaseObject.**Tracks**

The **Tracks** property syntax has the following parts.

Part	Description
CdBaseObject	The CdBase object
Integer	The number of tracks on the CD

Example

```
Private Sub Command1_Click()  
    Dim I As Integer  
    Dim SongList As String  
    Dim CddbResponse As String  
  
    SongList = ""  
    CddbResponse = CdBase1.GetCddb  
    For I = 1 to CdBase1.Tracks  
        SongList = SongList + CdBase1.SongName(I) + vbCrLf  
    Next I  
    MsgBox Songlist  
    CdBase1.SongName(I)
```

UseFirstMatch Property

Returns or sets the whether the control uses the first close match or pops up a choice windows

Syntax

Object.UseFirstMatch = Boolean

The **UseFirstMatch** property syntax has these parts:

Part	Description
<i>Object</i>	The CdBase object.
<i>Boolean</i>	Whether or not the control uses the first close match Default False (pops up window)

Data Type

Boolean

Remarks

Many programs simply read the CD information from the first close match instead of presenting a list of choices. This property allow use to go that route.

Visible Property

Determines if the CdBase control is visible or invisible at run time.

Syntax

Object.Visible = Boolean

The **Visible** property syntax has the following Parts

Part	Description
<i>Object</i>	The CdBase object
<i>Boolean</i>	True = Visible False = Hidden

cddbSend Method

Sends a command string to the cddb server

Syntax

Object.cddbSend(*SendString* As String)

The **cddbSend** method syntax has these parts:

Part	Description
<i>Object</i>	The CdBase object
<i>SendString</i>	The command string to be send.

Remarks

This is used for communications with the server. It is generally not needed unless you want your application to communicate with the server instead of the CdBase control.

See Cddb Server Command Reference for valid commands

Example

Coming soon

GetCategory Method

Returns a string representing the category name of the requested category number.

Syntax

CdBaseObject.**GetCategory**(CategoryNo As Integer) As String

The **GetCategory** method syntax has the following parts

Part	Description
------	-------------

<i>CdBaseObject</i>	The CdBase object.
---------------------	--------------------

<i>CategoryNo</i>	The number of the requested category.
-------------------	---------------------------------------

Remarks

This method should be called after the cddbiscat method is called. If the Category number is outside the range of available categories, the value returned will be an empty string.

```
Private Sub mnuCategory_Click()  
    Dim I As Integer  
    Dim NumCategories As Integer  
    Dim CatList As String  
    CatList = ""  
    NumCategories = CdBase1.Getlscat  
    For I = 1 To NumCategories  
        CatList = CatList + CdBase1.GetCategory(I) + vbCrLf  
    Next I  
    MsgBox CatList, vbInformation, "Category List"  
End Sub
```

GetCddb Method

Return a string representing a semi decoded response to a cddb read command

Syntax

Object.**GetCddb** As String

The **GetCddb** method syntax has the following parts;

Part	Description
<i>Object</i>	The CdBase object
<i>String</i>	A multiline String of Songs

Remarks

A request to the server is done each time the method is called. The multi-line string is in the following format. While the returned string is somewhat useful, there are other, easier ways to retrieve the information.

```
Joe / Joe's CD
3
Song 1
Song 2
Song 3
```

Example

```
Private Sub Command1_Click
    Dim cddbRead As String
    CddbRead = Getcddb
    MsgBox cddbRead
End Sub
```

GetCdDriveLetters Method

Returns an integer representing the number of CdDrives found

Syntax

Object.**GetCdDriveLetters**(Drives() As Variant) As Integer

The GetCdDriveLetters method has the following parts

Part	Description
<i>Object</i>	The CdBase object
<i>Drives()</i>	An undimensioned array of Variants

Remarks

Example

```
Private Sub Command1_Click()  
    Dim DriveLetters() As Variant  
    Dim NumDrives As Integer  
    Dim I As Integer  
    NumDrives = CdBase1.GetCdDriveLetters(DriveLetters())  
    For I = 1 To NumDrives  
        MsgBox DriveLetters(I)  
    Next 'I  
End Sub
```

GetCddbUseQueryString Method

Return a string representing a semi decoded response to a cddb read command

Syntax

Object.**GetCddbUseQueryString**(QueryString As String) As String

The **GetCddbUseQueryString** method syntax has the following parts;

Part	Description
<i>Object</i>	The CdBase object
QueryString	A string created by the <u>GetQueryString</u>
<i>String</i>	A multiline String of Songs

Remarks

A request to the server is done each time the method is called. The multi-line string is in the following format. While the returned string is somewhat useful, there are other, easier ways to retrieve the information.

Joe / Joe's CD
3
Song 1
Song 2
Song 3

Example

GetExtTrackData Method

Returns a string representing the extended track data for a selected track

Syntax

Object.**GetExtTrackData**(*SongNumber* As Integer) As String

The **GetExtTrackData** method syntax has the following parts

Part	Description
<i>Object</i>	The CdBase object
<i>SongNumber</i>	An Integer representing the song number

Remarks

Example

Getlscat Method

Returns an integer representing the number of categories received from the cddb server.

Syntax

Object.**Getlscat** As Integer

The **Getlscat** method syntax has the following parts

Part	Description
-------------	--------------------

<i>Object</i>	The CdBase object.
---------------	--------------------

<i>Integer</i>	The number of returned categories
----------------	-----------------------------------

Remarks

Example

```
Private Sub Command1_Click()  
    Dim I As Integer  
    Dim NumCategories As Integer  
    Dim CatList as String  
    NumCategories = Getlscat  
        For I = 1 To NumCategories  
            CatList = CatList + CdBase1.GetCategory(I) + vbCrLf  
        Next I  
    MsgBox CatList  
End Sub
```

GetMotd Method

Returns a string representing the message of the day from the server

Syntax

Object.**GetMotd** As String

The **GetMotd** method syntax has the following parts

Part	Description
<i>Object</i>	The Cdbase object.
<i>String</i>	The Message of the day

Remarks

Example

```
Private Sub Command1_Click()  
    MsgBox = GetMotd  
End Sub
```

GetMSid

Syntax

Object.**GetMSid** As String

The GetMSid method has the following parts

Part	Description
<i>Object</i>	The CdBase object.
<i>String</i>	The MSid

Remarks

This is the id that the Microsoft cdplayer application uses to identify CDs. This can be used by the programmer to compare and convert data. Microsoft stores the information in cdplayer.ini

Example

```
Private Sub Command1_Click()  
    MsgBox = GetMSid("D")  
End Sub
```

GetMyHostName Method

Returns a string representing the HostName of the client computer.

Syntax

Object.**GetMyHostName** As String

The **GetMyHostName** method syntax has the following parts

Part	Description
<i>Object</i>	The CdBase object.
<i>String</i>	The hostname

Remarks

This can be used to set the hostName property, but it should be a suggestion only because it has returned odd things on some computers.

Example

```
Private Sub Command1_Click()  
    TxtMyName.Text = CdBase1.GetMyHostName  
End Sub
```

GetQueryString Method

Returns a string that can be used with the [GetCddbUseQueryString](#) Method

Syntax

CdBase1.GetQueryString() As String

The **GetQueryString** method syntax has the following parts

Part	Description
<i>Object</i>	The CdBase object

Remarks

The dirty little secret of cddb is that it takes more than just the ID to do a query. This function allows query strings to be generated and stored for later bulk querying the server.

Example

```
TheQueryString = CdBase1.GetQueryString
MsgBox "Below is the query string for the current CD" + vbCrLf + vbCrLf _
      + TheQueryString + vbCrLf + vbCrLf + "Hit OK to access database with this string"
CddbResponse = CdBase1.GetCddbUseQueryString(TheQueryString)
```

GetSites Method

Returns an integer representing the total number of sites returned after retrieving the list of current cddb sites.

Syntax

Object.**GetSites** As Integer

The GetSites method syntax has these parts:

Part	Description
<i>Object</i>	The CdBase object

Remarks

This method queries the server for the available sites.

Example

```
Private Sub Command1_Click()  
    Dim I As Integer  
    Dim NumSites As Integer  
    Dim SiteList as String  
    NumSites = CdBase1.GetSites  
    NumSites = CdBase1.Sites 'also returns sites  
    For I = 1 To NumSites  
        SiteList = SiteList + CdBase1.Site(I) + vbCrLf  
    Next I  
    MsgBox SiteList  
End Sub
```

NewLineFix Method

Returns a string with all \n 's converted to vbCrLf 's

Syntax

Object.**NewLineFix**(*InputString*) As *String*

The NewLineFix method syntax has these parts:

Part	Description
<i>Object</i>	The CdBase object
<i>InputString</i>	The string to converted
<i>String</i>	The converted String

Remarks

This will convert strings like “The Best of Bob\nLive” to a two line string

Example

PlayTrack Method

Plays the indicated track

Syntax

Object.**PlayTrack**(TrackNumber As Integer)

The **PlayTrack** method syntax has these parts:

Part	Description
Object	The CdBase object
TrackNumber	The number of the track to be played

Remarks

The CD will begin playing at the track specified. If the track does not exist, the CD player will ignore the command.

Example

```
Private Sub Command1_Click()  
    CdBase1.PlayTrack (Val (InputBox("Song Number", "Play Track")))  
End Sub
```

Position Method

Returns the current position of the CD player relative to the start of the first track.

Syntax

Object.**Position**(TimeFormat As Integer) as string

The **Position** method syntax has these parts:

Part	Description
-------------	--------------------

<i>Object</i>	The CdBase object
<i>TimeFormat</i>	The format of the string returned.

Remarks

TimeFormat	Description
-------------------	--------------------

1	MSF (Minutes, Seconds, Frames)
2	Frames. The time in frames (75 frames = 1 second)
3	Seconds. The time in seconds
3	Clock. This return the position in digital clock format. # minutes 5 seconds would be "3:05"

This can be useful for displaying the current time. If this method is associated with a timer event a running counter is simulated.

Example

```
Private Sub Timer1_Timer()  
    Form1.Caption = "CD Position: " + CdBase1.Position(4)  
End Sub
```

SetExtTrackData Method

Sets the extended track data for a selected track

Syntax

Object.**SetExtTrackData**(*SongNumber* As Integer, *SetData* As String)

The **SetExtTrackData** method syntax has the following parts:

Part	Description
<i>Object</i>	The CdBase object
<i>SongNumber</i>	The number of the track
<i>SetData</i>	The extended data for the track

Remarks

This information should be set before submitting it to the server. Extended track data is optional. It is not required by the cddb server.

Example

SetSongName

Sets the song name for a selected track

Syntax

Object.**SetSongName**(*SongNumber* As Integer, *SongName* As String)

The **SetSongName** method syntax has the following parts:

Part	Description
<i>Object</i>	The CdBase object
<i>SongNumber</i>	The number of the track
<i>SetName</i>	The song name for a selected track

Remarks

This information should be set before submitting it to the server.

Example

SiteAddress Method

Returns a String representing address of the selected cddb server site

Syntax

Object.**SiteAddress**(SiteNumber As Integer) As String

The **SiteAddress** method syntax has the following parts:

Part	Description
<i>Object</i>	The CdBase object
<i>SiteNumber</i>	The number of the site in the list

Remarks

Example

```
Private Sub Command1_Click()  
    Dim NumSites As Integer  
    NumSites = CdBase1.GetSites  
    MsgBox CdBase1.SiteAddress(1)  
End Sub
```

SiteLocation Method

Returns a String representing location of the selected cddb server site

Syntax

Object.**SiteLocation**(SiteNumber As Integer) As String

The **SiteLocation** method syntax has the following parts:

Part	Description
<i>Object</i>	The CdBase object
<i>SiteNumber</i>	The number of the site in the list

Remarks

Example

```
Private Sub Command1_Click()  
    Dim NumSites As Integer  
    NumSites = CdBase1.GetSites  
    MsgBox CdBase1.SiteLocation(1)  
End Sub
```

SitePort Method

Returns an Integer representing port number of the selected cddb server site

Syntax

Object.**SitePort**(SiteNumber As Integer) As Integer

The SitePort method syntax has the following parts:

Part	Description
<i>Object</i>	The CdBase object
<i>SiteNumber</i>	The number of the site in the list

Remarks

Example

```
Private Sub Command1_Click()  
    Dim NumSites As Integer  
    NumSites = CdBase1.GetSites  
    MsgBox CdBase1.SitePort(1)  
End Sub
```

SongLength Method

Returns a long representing the length of the specified song in seconds.

Syntax

Object.**SongLength**(SongNumber As Integer , FromCD As Boolean) As Long

The **SongLength** method syntax has the following parts

Part	Description
<i>Object</i>	The CdBase object
<i>SongNumber</i>	An Integer representing the song number
<i>FromCD</i>	True = Read from CD False = Read from cddb database
<i>Long</i>	Length of CD in seconds

Remarks

This returns the length of song in seconds. If the song number is out of the range of valid songs, the value returned will be zero. If FromCD = False then length is given from information retrieved from database. The [GetCddb](#) method must be called first. The **SongLength** method will not query the server.

Example

```
Private Sub Command1_Click()  
MsgBox "Song 1 is " + Cstr(CdBase1.SongLength(1)) + "seconds"  
End sub
```


SongName Method

Returns a string representing the name of the requested song.

Syntax

Object.**SongName**(SongNumber As Integer) As String

The **SongName** method syntax has the following parts

Part	Description
Object	The CdBase object
SongNumber	An Integer representing the song number

Remarks

This returns the name of song. If the song number is out of the range of valid songs, the value returned will be an empty string

Example

```
Private Sub Command1_Click()  
    MsgBox "Song 1 name is " + CdBase1.SongName(1)  
End sub
```


SubmitString Method

Returns a string fully formatted for a cddb submission

Syntax

Object.SubmitString() as *String*

The SubmitString method syntax has the following parts

Part	Description
<i>Object</i>	The CdBase Object
<i>String</i>	The formatted string

Remarks

This string can be emailed to the server with the subject in the following format 'cddb categ Cdid'
Example: cddb rock 850f970b. You can test it by sending it to cddb-test@submit.cddb.com

Example

DataSent Event

Occurs when data from the client or server is sent.

Syntax

object_ **DataSent** (bytesTotal As Long , Data As String)

The **DataSent** event syntax has these parts:

Part	Description
object	the CdBase object.
bytesTotal	Long. The total amount of data that can be retrieved.
Data	String. The data being sent.

Remarks

This event is used if you wish to log the communications being sent back and forth between the client and the server. If you do not wish to log the data you can ignore the event.

Example

```
Private Sub CdBase1_DataSent(bytesTotal As Long, Data As String)
    Text1.Multiline = True
    Text1.Text = Text1.Text + Data
    Text1.SelStart = Len(Text2.Text) 'keeps text scrolling
End Sub
```


ModeChange Event

Occurs when the mode of the CD player has changed

Syntax

Object **ModeChange**(*NewMode* as String)

The **ModeChange** Event syntax has the following parts:

Part	Description
-------------	--------------------

<i>Object</i>	The CdBase object
<i>NewMode</i>	A string describing the CD mode

Remarks

Example

```
Private Sub CdBase1_ModeChange(NewMode as String)
    Text.Text = "New Mode: " + NewMode
End Sub
```

CDDDB Server Commands (edited from the cddb faq file)

