



PUBLIC

SQL Anywhere - MobiLink

Document Version: 17.01.0 – 2021-10-15

MobiLink - Dbmlsync .NET API Reference

Content

- 1 Dbmsync .NET API Reference. 3**
- 1.1 DbmsyncClient Class. 5
 - CancelSync Method. 7
 - Connect(String, Int32, String, String) Method. 10
 - Disconnect() Method. 11
 - Fini() Method. 12
 - GetErrorInfo() Method. 13
 - GetEvent(out DBSC_Event, UInt32) Method. 13
 - GetProperty(String, out String) Method. 15
 - Init() Method. 16
 - InstantiateClient() Method. 16
 - Ping(UInt32) Method. 17
 - SetProperty(String, String) Method. 18
 - ShutdownServer(DBSC_ShutdownType) Method. 20
 - StartServer(Int32, String, UInt32, out DBSC_StartType) Method. 21
 - Sync(String, String) Method. 22
 - WaitForServerShutdown(UInt32) Method. 23
- 1.2 DBSC_CancelRet Enumeration. 24
- 1.3 DBSC_ErrorType Enumeration. 25
- 1.4 DBSC_EventType Enumeration. 27
- 1.5 DBSC_GetEventRet Enumeration. 29
- 1.6 DBSC_ShutdownType Enumeration. 30
- 1.7 DBSC_StartType Enumeration. 31
- 1.8 DBSC_ErrorInfo Structure. 32
- 1.9 DBSC_Event Structure. 33

1 Dbmsync .NET API Reference

The Dbmsync .NET API provides a programming interface that allows MobiLink client applications written in .NET to launch synchronizations and receive feedback about the progress of the synchronizations they request.

Namespace

```
Sap.MobiLink.Client
```

The dbmsync .NET API reference is available in the *MobiLink - Dbmsync .NET API Reference* at <https://help.sap.com/viewer/63256cabfb224813ac3854b94c7f0bdf/LATEST/en-US>.

Example

The sample below shows a typical application using the .NET version of the Dbmsync API to perform a synchronization to receive output events. The sample omits error handling for clarity. It is always good indent practice to check the return value from each API call.

```
using System;
using System.Collections.Generic;
using System.Text;
using Sap.MobiLink.Client;
namespace ConsoleApplication6
{
    class Program
    {
        static void Main(string[] args)
        {
            DbmsyncClient cli1;
            DBSC_StartType st1;
            DBSC_Event ev1;
            UInt32 syncHdl;
            cli1 = DbmsyncClient.InstantiateClient();
            cli1.Init();
            // Setting the server path is usually not required unless
            // your SQL Anywhere install is not in your path or you have multiple
            // versions of the product installed.
            cli1.SetProperty("server path", "d:\\sap\\sqlany17\\bin32;");
            cli1.StartServer(3426,
                "-c server=cons;dbn=rem1;uid=dba;pwd=passwd -ve+ -ot c:\\temp\\
            \\dbsync1.txt",
                5000, out st1);
            cli1.Connect(null, 3426, "dba", "sql");
            syncHdl = cli1.Sync("spl", "");
            while (cli1.GetEvent(out ev1, 5000)
                == DBSC_GetEventRet.DBSC_GETEVENT_OK)
            {
                if (ev1.hdl == syncHdl)
                {
```

```

        Console.WriteLine("Event Type : {0}", ev1.type);
        if (ev1.type == DBSC_EventType.DBSC_EVENTTYPE_INFO_MSG)
        {
            Console.WriteLine("Info : {0}", ev1.str1);
        }
        if (ev1.type == DBSC_EventType.DBSC_EVENTTYPE_SYNC_DONE)
        {
            break;
        }
    }
}
cli1.ShutdownServer(DBSC_ShutdownType.DBSC_SHUTDOWN_ON_EMPTY_QUEUE);
cli1.WaitForServerShutdown(10000);
cli1.Disconnect();
cli1.Fini();
Console.ReadLine();
    }
}
}

```

In this section:

[DbmlsyncClient Class \[page 5\]](#)

Communicates using TCP/IP with a separate process, dbmlsync server, which performs a synchronization by connecting to the MobiLink server and the remote database.

[DBSC_CancelRet Enumeration \[page 24\]](#)

Indicates the result of a synchronization cancellation attempt.

[DBSC_ErrorType Enumeration \[page 25\]](#)

Indicates the reason for a method call failure.

[DBSC_EventType Enumeration \[page 27\]](#)

Indicates the type of event generated by a synchronization.

[DBSC_GetEventRet Enumeration \[page 29\]](#)

Indicates the result of an attempt to retrieve an event.

[DBSC_ShutdownType Enumeration \[page 30\]](#)

Indicates how urgently the server should be shut down.

[DBSC_StartType Enumeration \[page 31\]](#)

Indicates the action taken during a dbmlsync server startup attempt.

[DBSC_ErrorInfo Structure \[page 32\]](#)

Contains information about the failure of a previous method call.

[DBSC_Event Structure \[page 33\]](#)

Contains information about an event generated by a synchronization.

1.1 DbmlsyncClient Class

Communicates using TCP/IP with a separate process, dbmlsync server, which performs a synchronization by connecting to the MobiLink server and the remote database.

Syntax

Visual Basic

```
Public Class DbmlsyncClient
```

C#

```
public class DbmlsyncClient
```

Members

All members of DbmlsyncClient, including inherited members.

Methods

Modifier and Type	Method	Description
public Boolean	CancelSync [page 7]	Cancels a synchronization request.
public Boolean	Connect(String, Int32, String, String) [page 10]	Opens a connection to a dbmlsync server that is already running on this computer.
public Boolean	Disconnect() [page 11]	Breaks the dbmlsync server connection that was established with the Connect method.
public Boolean	Fini() [page 12]	Frees all resources used by this class instance.
public DBSC_ErrorInfo	GetErrorInfo() [page 13]	Retrieves additional information about the failure after a DbmlsyncClient class method returns a failed return code.
public DBSC_GetEventRet	GetEvent(out DBSC_Event, UInt32) [page 13]	Retrieves the next feedback event for synchronizations requested by the client.
public Boolean	GetProperty(String, out String) [page 15]	Retrieves the current value of a property.
public Boolean	Init() [page 16]	Initializes a DbmlsyncClient class instance.
public static DbmlsyncClient	InstantiateClient() [page 16]	Creates an instance of the dbmlsync client class that can be used to control synchronizations.

Modifier and Type	Method	Description
public Boolean	Ping(UInt32) [page 17]	Sends a ping request to the dbmsync server to check if the server is active and responding to requests.
public Boolean	SetProperty(String, String) [page 18]	Sets various properties to modify the behavior of the class instance.
public Boolean	ShutdownServer(DBSC_Shutdown-Type) [page 20]	Shuts down the dbmsync server to which the client is connected.
public Boolean	StartServer(Int32, String, UInt32, out DBSC_StartType) [page 21]	Starts a new dbmsync server if one is not already listening on the specified port.
public UInt32	Sync(String, String) [page 22]	Requests that the dbmsync server perform a synchronization.
public Boolean	WaitForServerShutdown(UInt32) [page 23]	Returns when the server has shutdown or when the timeout expires, whichever comes first.

Remarks

Multiple clients can share the same dbmsync server. However, each dbmsync server can only synchronize a single remote database. Each remote database can have only one dbmsync server synchronizing it.

The dbmsync server performs one synchronization at a time. If the server receives a synchronization request while performing a synchronization, it queues that request and satisfies it later.

Status information generated by synchronizations is communicated back to the client application through the `GetEvent` method.

In this section:

[CancelSync Method \[page 7\]](#)

Cancels a synchronization request.

[Connect\(String, Int32, String, String\) Method \[page 10\]](#)

Opens a connection to a dbmsync server that is already running on this computer.

[Disconnect\(\) Method \[page 11\]](#)

Breaks the dbmsync server connection that was established with the `Connect` method.

[Fini\(\) Method \[page 12\]](#)

Frees all resources used by this class instance.

[GetErrorInfo\(\) Method \[page 13\]](#)

Retrieves additional information about the failure after a `DbmsyncClient` class method returns a failed return code.

[GetEvent\(out DBSC_Event, UInt32\) Method \[page 13\]](#)

Retrieves the next feedback event for synchronizations requested by the client.

[GetProperty\(String, out String\) Method \[page 15\]](#)

Retrieves the current value of a property.

[Init\(\) Method \[page 16\]](#)

Initializes a DbmlsyncClient class instance.

[InstantiateClient\(\) Method \[page 16\]](#)

Creates an instance of the dbmlsync client class that can be used to control synchronizations.

[Ping\(UInt32\) Method \[page 17\]](#)

Sends a ping request to the dbmlsync server to check if the server is active and responding to requests.

[SetProperty\(String, String\) Method \[page 18\]](#)

Sets various properties to modify the behavior of the class instance.

[ShutdownServer\(DBSC_ShutdownType\) Method \[page 20\]](#)

Shuts down the dbmlsync server to which the client is connected.

[StartServer\(Int32, String, UInt32, out DBSC_StartType\) Method \[page 21\]](#)

Starts a new dbmlsync server if one is not already listening on the specified port.

[Sync\(String, String\) Method \[page 22\]](#)

Requests that the dbmlsync server perform a synchronization.

[WaitForServerShutdown\(UInt32\) Method \[page 23\]](#)

Returns when the server has shutdown or when the timeout expires, whichever comes first.

Related Information

[GetEvent\(out DBSC_Event, UInt32\) Method \[page 13\]](#)

1.1.1 CancelSync Method

Cancels a synchronization request.

Overload list

Modifier and Type	Overload name	Description
public Boolean	CancelSync(UInt32) [page 8]	Allows a client to cancel a synchronization request previously made using the Sync method.
public DBSC_CancelRet	CancelSync(UInt32, Boolean) [page 9]	Allows a client to cancel a synchronization request previously made using the Sync method.

In this section:

[CancelSync\(UInt32\) Method \(Deprecated\) \[page 8\]](#)

Allows a client to cancel a synchronization request previously made using the Sync method.

[CancelSync\(UInt32, Boolean\) Method \[page 9\]](#)

Allows a client to cancel a synchronization request previously made using the Sync method.

1.1.1.1 CancelSync(UInt32) Method (Deprecated)

Allows a client to cancel a synchronization request previously made using the Sync method.

Syntax

Visual Basic

```
Public Function CancelSync (ByVal hdl As UInt32) As Boolean
```

C#

```
public Boolean CancelSync (UInt32 hdl)
```

Parameters

hdl The synchronization handle returned by the Sync method when the synchronization was requested.

Returns

True when the synchronization request was successfully canceled; otherwise, returns false. When false is returned, you can call the GetErrorInfo method for more information about the failure.

Remarks

Only synchronization requests waiting to be serviced can be canceled. To stop a synchronization that has already begun, use the CancelSync(UInt32, Boolean) method.

A connection must be established to the server before this method can be used. This method cannot be used if the client has disconnected from the server since the Sync method was called.

Related Information

[GetErrorInfo\(\) Method \[page 13\]](#)

[CancelSync\(UInt32, Boolean\) Method \[page 9\]](#)

[ShutdownServer\(DBSC_ShutdownType\) Method \[page 20\]](#)

1.1.1.2 CancelSync(UInt32, Boolean) Method

Allows a client to cancel a synchronization request previously made using the Sync method.

Syntax

Visual Basic

```
Public Function CancelSync (  
    ByVal hdl As UInt32,  
    ByVal cancel_active As Boolean  
) As DBSC_CancelRet
```

C#

```
public DBSC_CancelRet CancelSync (  
    UInt32 hdl,  
    Boolean cancel_active  
)
```

Parameters

hdl The synchronization handle returned by the Sync method when the synchronization was requested.

cancel_active When set to true, the request is canceled even if the synchronization has already begun. When set to false, the request is only canceled if synchronization has not yet begun

Returns

A value from the DBSC_CancelRet enumeration. When DBSC_CANCEL_FAILED is returned, you can call the GetErrorInfo method for more information about the failure.

Remarks

A connection must be established to the server before this method can be used. This method cannot be used if the client has disconnected from the server since the Sync method was called.

Related Information

[DBSC_CancelRet Enumeration \[page 24\]](#)

[GetErrorInfo\(\) Method \[page 13\]](#)

[ShutdownServer\(DBSC_ShutdownType\) Method \[page 20\]](#)

1.1.2 Connect(String, Int32, String, String) Method

Opens a connection to a dbmlsync server that is already running on this computer.

☰ Syntax

Visual Basic

```
Public Function Connect (  
    ByVal host As String,  
    ByVal port As Int32,  
    ByVal uid As String,  
    ByVal pwd As String  
) As Boolean
```

C#

```
public Boolean Connect (  
    String host,  
    Int32 port,  
    String uid,  
    String pwd  
)
```

Parameters

host This value is reserved. Specify null when using C#. Do not specify anything when using Visual Basic.

port The TCP port on which the dbmlsync server is listening. Use the same port value that you specified with the StartServer method.

uid A valid database user id that has the SYS_AUTH_DBA_ROLE system role, a user-extended role based on server_userid, or be the same as server_userid.

pwd The database password for the user specified by uid.

Returns

True when a connection to the server was established; otherwise, returns false. When false is returned, you can call the GetErrorInfo method for more information about the failure.

Remarks

Dbmlsync servers are started (either using the command line or the StartServer method) using a connection string that provided a database userid and password (for example, *server_userid*). In addition, the Connect method of the Dbmlsync API requires a valid database userid (for example, *client_userid*).

Client_userid is used only to validate whether this client has sufficient permissions to synchronize the database. When synchronizations are performed *server_userid* is used.

In SQL Anywhere 12 and earlier, both *client_userid* and *server_userid* required DBA or REMOTE DBA permissions.

In SQL Anywhere 16 and later, *server_userid* must have sufficient privileges to synchronize. At a minimum, *server_userid* must have the SYS_RUN_REPLICATION_ROLE system role but other privileges may be required to allow synchronization. *Client_userid* must either:

- Be the same as *server_userid*, or
- Have the SYS_AUTH_DBA_ROLE system role, or
- Have a user-extended role based on *server_userid*, for example:CREATE ROLE FOR USER server_userid;GRANT server_userid to client_userid;

While the last option ensures that *client_userid* has at least as many system privileges as *server_userid*, it is also possible for *server_userid* to have object-level privileges not granted to its role and therefore *client_userid* would not have those privileges. If those privileges are used during synchronization, then *client_userid* has effectively increased its privileges to perform the synchronization. If this is not acceptable, ensure that all of *server_userid*'s object-level privileges are granted to its user-extended role.

Related Information

[StartServer\(Int32, String, UInt32, out DBSC_StartType\) Method \[page 21\]](#)

[GetErrorInfo\(\) Method \[page 13\]](#)

1.1.3 Disconnect() Method

Breaks the dbmsync server connection that was established with the Connect method.

☰ Syntax

Visual Basic

```
Public Function Disconnect () As Boolean
```

C#

```
public Boolean Disconnect ()
```

Returns

True when the connection to the server has been broken; otherwise, returns false. When false is returned, you can call the GetErrorInfo method for more information about the failure.

Remarks

You should always call `Disconnect` when you are finished with a connection.

Related Information

[Connect\(String, Int32, String, String\) Method \[page 10\]](#)

[GetErrorInfo\(\) Method \[page 13\]](#)

1.1.4 Fini() Method

Frees all resources used by this class instance.

≡ Syntax

Visual Basic

```
Public Function Fini () As Boolean
```

C#

```
public Boolean Fini ()
```

Returns

True when the class instance is successfully finalized; otherwise, returns false. When false is returned, you can call the `GetErrorInfo` method for more information about the failure.

Remarks

This method must be called before you can delete the `DbmlSyncClient` class instance.

i Note

You should use the `Disconnect` method to disconnect from any connected servers before finalizing the class instance.

Related Information

[Disconnect\(\) Method \[page 11\]](#)

[GetErrorInfo\(\) Method \[page 13\]](#)

1.1.5 GetErrorInfo() Method

Retrieves additional information about the failure after a DbmlsyncClient class method returns a failed return code.

Syntax

Visual Basic

```
Public Function GetErrorInfo () As DBSC_ErrorInfo
```

C#

```
public DBSC_ErrorInfo GetErrorInfo ()
```

Returns

A pointer to a DBSC_ErrorInfo structure that contains information about the failure. The contents of this structure may be overwritten the next time any class method is called.

Related Information

[DBSC_ErrorType Enumeration \[page 25\]](#)

[DBSC_ErrorInfo Structure \[page 32\]](#)

1.1.6 GetEvent(out DBSC_Event, UInt32) Method

Retrieves the next feedback event for synchronizations requested by the client.

Syntax

Visual Basic

```
Public Function GetEvent (  
    ByVal ev As DBSC_Event,  
    ByVal timeout As UInt32  
) As DBSC_GetEventRet
```

C#

```
public DBSC_GetEventRet GetEvent (
    out DBSC_Event ev,
    UInt32 timeout
)
```

Parameters

ev If the return value is DBSC_GETEVENT_OK then the ev parameter is filled with information about the event that has been retrieved.

timeout Indicates the maximum time in milliseconds to wait if no event is immediately available to return. Use DbmlsyncClient.DBSC_INFINITY to wait indefinitely for a response.

Returns

A value from the DBSC_GetEventRet enumeration. When DBSC_GETEVENT_FAILED is returned, you can call the GetErrorInfo method for more information about the failure.

Remarks

Feedback events contain information such as messages generated from the sync, data for updating a progress bar, and synchronization cycle notifications.

As the dbmlsync server runs a synchronization it generates a series of events that contain information about the progress of the synchronization. These events are sent from the server to the DbmlsyncClient class, which queues them. When the GetEvent method is called, the next event in the queue is returned if there is one waiting.

If there are no events waiting in the queue, this method waits until an event is available or until the specified timeout has expired before returning.

The types of events that are generated for a synchronization can be controlled using properties.

Related Information

[DBSC_GetEventRet Enumeration \[page 29\]](#)

[DBSC_Event Structure \[page 33\]](#)

[SetProperty\(String, String\) Method \[page 18\]](#)

[GetErrorInfo\(\) Method \[page 13\]](#)

1.1.7 GetProperty(String, out String) Method

Retrieves the current value of a property.

☰ Syntax

Visual Basic

```
Public Function GetProperty (  
    ByVal name As String,  
    ByVal value As String  
) As Boolean
```

C#

```
public Boolean GetProperty (  
    String name,  
    out String value  
)
```

Parameters

name The name of the property to retrieve. For a list of valid property names, see SetProperty.

value On exit, the value of the property is stored in this variable.

Returns

True when the property was successfully received; otherwise, returns false. When false is returned, you can call the GetErrorInfo method for more information about the failure.

Related Information

[SetProperty\(String, String\) Method \[page 18\]](#)

[GetErrorInfo\(\) Method \[page 13\]](#)

1.1.8 Init() Method

Initializes a DbmlsyncClient class instance.

☰ Syntax

Visual Basic

```
Public Function Init () As Boolean
```

C#

```
public Boolean Init ()
```

Returns

True when the class instance is successfully initialized; otherwise, returns false. When false is returned, you can call the GetErrorInfo method for more information about the failure.

Remarks

This method must be called after instantiating the DbmlSyncClient class instance. Other DbmlSyncClient methods cannot be called until you have successfully initialized the instance.

Related Information

[InstantiateClient\(\) Method \[page 16\]](#)

[GetErrorInfo\(\) Method \[page 13\]](#)

1.1.9 InstantiateClient() Method

Creates an instance of the dbmlsync client class that can be used to control synchronizations.

☰ Syntax

Visual Basic

```
Public Shared Function InstantiateClient () As DbmlsyncClient
```

C#

```
public static DbmlsyncClient InstantiateClient ()
```


Returns

The created DbmlsyncClient instance. Returns null when an error occurs.

Remarks

The object returned by this method can be used to call the remaining methods in the class.

1.1.10 Ping(UInt32) Method

Sends a ping request to the dbmlsync server to check if the server is active and responding to requests.

Syntax

Visual Basic

```
Public Function Ping (ByVal timeout As UInt32) As Boolean
```

C#

```
public Boolean Ping (UInt32 timeout)
```

Parameters

timeout The maximum number of milliseconds to wait for the server to respond to the ping request. Use DbmlsyncClient.DBSC_INFINITY to wait indefinitely for a response.

Returns

True when a response to the ping request was received from the server; otherwise, returns false. When false is returned, you can call the GetErrorInfo method for more information about the failure.

Remarks

You must be connected to the server before calling this method.

Related Information

[GetErrorInfo\(\) Method \[page 13\]](#)

1.1.11 SetProperty(String, String) Method

Sets various properties to modify the behavior of the class instance.

Syntax

Visual Basic

```
Public Function SetProperty (
    ByVal name As String,
    ByVal value As String
) As Boolean
```

C#

```
public Boolean SetProperty (
    String name,
    String value
)
```

Parameters

name The name of the property to set. For a list of valid property names, see table.

value The value to set for the property.

Returns

True when the property was successfully set; otherwise, returns false. When false is returned, you can call the `GetErrorInfo` method for more information about the failure.

Remarks

Changes to property values only affect synchronization requests made after the property value was changed.

The *server path* property can be set to specify the directory from which the client should start `dbmlsync.exe` when the `StartServer` method is called. When this property is not set, `dbmlsync.exe` is found using the `PATH` environment variable. If there are multiple versions of SQL Anywhere installed on your computer, it is recommended that you specify the location of `dbmlsync.exe` using the *server path* property because the `PATH`

environment variable may locate a dbmsync executable from another installed version of SQL Anywhere. For example,

```
ret = cli->SetProperty("server path", "c:\\sa12\\bin32");
```

The properties control the types of events that are returned by the GetEvent method. By disabling events that you do not require you may be able to improve performance. An event type is enabled by setting the corresponding property to "1" and disabled by setting the property to "0".

The following is a table of available property names and the event types that each name controls:

Property name	Event types controlled	Default value
enable errors	DBSC_EVENTTYPE_ERROR_MSG	1
enable warnings	DBSC_EVENTTYPE_WARNING_MSG	1
enable info msgs	DBSC_EVENTTYPE_INFO_MSG	1
enable progress	DBSC_EVENTTYPE_PROGRESS_IN- DEX	0
enable progress text	DBSC_EVENTTYPE_PROGRESS_TEXT	0
enable title	DBSC_EVENTTYPE_TITLE	0
enable sync start	DBSC_EVENTTYPE_SYNC_START	1
enable sync done	DBSC_EVENTTYPE_SYNC_DONE	1
enable sync start and done	DBSC_EVENTTYPE_SYNC_START DBSC_EVENTTYPE_SYNC_DONE	1
enable status	DBSC_EVENTTYPE_ML_CONNECT DBSC_EVENTTYPE_UPLOAD_START DBSC_EVENTTYPE_UPLOAD_SENT DBSC_EVENTTYPE_UPLOAD_COM- MITTED DBSC_EVENTTYPE_DOWN- LOAD_START DBSC_EVENTTYPE_DOWN- LOAD_COMMITTED	1

Related Information

[StartServer\(Int32, String, UInt32, out DBSC_StartType\) Method \[page 21\]](#)

[GetEvent\(out DBSC_Event, UInt32\) Method \[page 13\]](#)

[GetProperty\(String, out String\) Method \[page 15\]](#)

[GetErrorInfo\(\) Method \[page 13\]](#)

1.1.12 ShutdownServer(DBSC_ShutdownType) Method

Shuts down the dbmlsync server to which the client is connected.

☰ Syntax

Visual Basic

```
Public Function ShutdownServer (ByVal how As DBSC_ShutdownType) As Boolean
```

C#

```
public Boolean ShutdownServer (DBSC_ShutdownType how)
```

Parameters

how Indicates the urgency of the server shutdown. Supported values are listed in the DBSC_ShutdownType enumeration.

Returns

True when a shutdown request was successfully sent to the server; otherwise, returns false. When false is returned, you can call the GetErrorInfo method for more information about the failure.

Remarks

The Shutdown method returns immediately but there may be some delay before the server actually shuts down.

The WaitForServerShutdown method can be used to wait until the server actually shuts down.

i Note

You should still use the Disconnect method after calling ShutdownServer.

Related Information

[DBSC_ShutdownType Enumeration \[page 30\]](#)

[Disconnect\(\) Method \[page 11\]](#)

[WaitForServerShutdown\(UInt32\) Method \[page 23\]](#)

[GetErrorInfo\(\) Method \[page 13\]](#)

1.1.13 StartServer(Int32, String, UInt32, out DBSC_StartType) Method

Starts a new dbmlsync server if one is not already listening on the specified port.

Syntax

Visual Basic

```
Public Function StartServer (  
    ByVal port As Int32,  
    ByVal cmdline As String,  
    ByVal timeout As UInt32,  
    ByVal starttype As DBSC_StartType  
    ) As Boolean
```

C#

```
public Boolean StartServer (  
    Int32 port,  
    String cmdline,  
    UInt32 timeout,  
    out DBSC_StartType starttype  
    )
```

Parameters

port The TCP port to check for an existing dbmlsync server. If a new server is started, it is set to listen on this port.

cmdline A valid command line for starting a dbmlsync server. The command line may contain only the following options which have the same meaning that they do for the dbmlsync utility: -a, -c, -dl, -do, -ek, -ep, -k, -l, -o, -os, -ot, -p, -pc+, -pc-, -pd, -pp, -q, -qi, -qc, -sc, -sp, -uc, -ud, -ui, -um, -un, -ux, -v[cnoprst], -wc, -wh. The -c option must be specified.

timeout The maximum time in milliseconds to wait after a dbmlsync server is started for it to be ready to accept requests. Use DbmlsyncClient.DBSC_INFINITY to wait indefinitely for a response.

starttype An out parameter set to indicate if the server has been located or started. If starttype is non-null on entry and StartServer returns true, then, on exit, the variable pointed to by starttype is set to a value from the DBSC_StartType enumeration.

Returns

True when the server was already running or successfully started; otherwise, returns false. When false is returned, you can call the GetErrorInfo method for more information about the failure.

Remarks

If a server is present, this method sets the starttype parameter to DBSC_SS_ALREADY_RUNNING and returns without further action. If no server is found, the method starts a new server using the options specified by the cmdline argument and waits for it to start accepting requests before returning.

On Windows Mobile devices, it is usually necessary to set the *server path* property before StartServer can be successfully called. The *server path* property does not need to be set in the following instances:

- Your application is in the same directory as dbmsync.exe.
- dbmsync.exe is in the Windows directory.

Related Information

[GetErrorInfo\(\) Method \[page 13\]](#)

1.1.14 Sync(String, String) Method

Requests that the dbmsync server perform a synchronization.

☰ Syntax

Visual Basic

```
Public Function Sync (
    ByVal syncName As String,
    ByVal opts As String
) As UInt32
```

C#

```
public UInt32 Sync (
    String syncName,
    String opts
)
```

Parameters

syncName The name of a synchronization profile defined in the remote database that contains the options for the synchronization. If syncName is null then no profile is used and the opts parameter should contain all the options for the synchronization.

opts A string formed according to the same rules used to define an option string for a synchronization profile, which is a string specified as a semicolon delimited list of elements of the form <option name>=<option value>. If syncName is non-null then the options specified by opts are added to those already in the synchronization profile specified by syncName. If an option in the string already exists in the

profile, then the value from the string replaces the value already stored in the profile. If the syncName is null then opts should specify all the options for the synchronization.

Returns

An integer value which uniquely identifies this synchronization request and is only valid until the client disconnects from the server. Returns NULL_SYNCHDL if an error prevents the synchronization request from being created. When NULL_SYNCHDL is returned, you can call the GetErrorInfo method for more information about the failure.

Remarks

You must be connected to the server before calling this method. At least one of syncName and opts must be non-null.

The return value identifies the synchronization request and can be used to cancel the request or to process events returned by the synchronization.

The following C# example demonstrates how to display error codes after invoking the Sync method.

```
// Insert code to initialize the synchronization client.
UInt32 request = syncClient.Sync("syncName", null);
if (request == DbmlsyncClient.NULL_SYNCHDL) {
    string error_code = syncClient.GetErrorInfo().type.ToString();
    MessageBox.Show(error_code, "Sync Error");
}
```

Related Information

[GetErrorInfo\(\) Method \[page 13\]](#)

1.1.15 WaitForServerShutdown(UInt32) Method

Returns when the server has shutdown or when the timeout expires, whichever comes first.

☰ Syntax

Visual Basic

```
Public Function WaitForServerShutdown (ByVal timeout As UInt32) As Boolean
```

C#

```
public Boolean WaitForServerShutdown (UInt32 timeout)
```

Parameters

timeout Indicates the maximum time in milliseconds to wait for the server to shutdown. Use DbmlsyncClient.DBSC_INFINITY to wait indefinitely for a response.

Returns

True when the method returned due to the server shutdown; otherwise, returns false. When false is returned, you can call the GetErrorInfo method for more information about the failure.

Remarks

WaitForServerShutdown can only be called after the ShutdownServer method is called.

Related Information

[GetErrorInfo\(\) Method \[page 13\]](#)

1.2 DBSC_CancelRet Enumeration

Indicates the result of a synchronization cancellation attempt.

≡ Syntax

Visual Basic

```
Public Enum DBSC_CancelRet
```

C#

```
enum DBSC_CancelRet
```


Members

Member name	Description	Value
DBSC_CANCEL_OK_QUEUED	Canceled a synchronization that was in the wait queue.	1
DBSC_CANCEL_OK_ACTIVE	Canceled an active synchronization.	2
DBSC_CANCEL_FAILED	Failed to cancel the synchronization.	3

Related Information

[CancelSync\(UInt32, Boolean\) Method \[page 9\]](#)

1.3 DBSC_ErrorType Enumeration

Indicates the reason for a method call failure.

↳ Syntax

Visual Basic

```
Public Enum DBSC_ErrorType
```

C#

```
enum DBSC_ErrorType
```

Members

Member name	Description	Value
DBSC_ERR_OK	No error occurred.	1
DBSC_ERR_NOT_INITIALIZED	The class has not been initialized by calling the Init method.	2
DBSC_ERR_ALREADY_INITIALIZED	The Init method was called on a class that was already initialized.	3
DBSC_ERR_NOT_CONNECTED	No connection to a dbmsync server is in place.	4
DBSC_ERR_CANT_RESOLVE_HOST	Cannot resolve host information.	5

Member name	Description	Value
DBSC_ERR_CONNECT_FAILED	Connection to the dbmsync server has failed.	6
DBSC_ERR_INITIALIZING_TCP_LAYER	Error initializing TCP layer.	7
DBSC_ERR_ALREADY_CONNECTED	Connect method failed because a connection was already in place.	8
DBSC_ERR_PROTOCOL_ERROR	This is an internal error.	9
DBSC_ERR_CONNECTION_REJECTED	The connection was rejected by the dbmsync server. str1 points to a string returned by the server which may provide more information about why the connection attempt was rejected.	10
DBSC_ERR_TIMED_OUT	The timeout expired while waiting for a response from the server.	11
DBSC_ERR_STILL_CONNECTED	Could not Fini the class because it is still connected to the server.	12
DBSC_ERR_SYNC_NOT_CANCELED	The server could not cancel the synchronization request, likely because the synchronization was already in progress.	14
DBSC_ERR_INVALID_VALUE	An invalid property value was passed to the SetProperty method.	15
DBSC_ERR_INVALID_PROP_NAME	The specified property name is not valid.	16
DBSC_ERR_VALUE_TOO_LONG	The property value is too long; properties must be less than DBCS_MAX_PROPERTY_LEN bytes long.	17
DBSC_ERR_SERVER_SIDE_ERROR	A server-side error occurred while canceling or adding a sync. str1 points to a string returned by the server which may provide more information about the error.	18
DBSC_ERR_CREATE_PROCESS_FAILED	Unable to start a new dbmsync server.	20
DBSC_ERR_READ_FAILED	TCP error occurred while reading data from the dbmsync server.	21
DBSC_ERR_WRITE_FAILED	TCP error occurred while sending data to the dbmsync server.	22
DBSC_ERR_NO_SERVER_RESPONSE	Failed to receive a response from the server that is required to complete the requested action.	23
DBSC_ERR_UID_OR_PWD_TOO_LONG	The UID or PWD specified is too long.	24

Member name	Description	Value
DBSC_ERR_UID_OR_PWD_NOT_VALID	The UID or PWD specified is not valid or lacks sufficient privileges to connect.	25
DBSC_ERR_INVALID_PARAMETER	One of the parameters passed to the function was not valid.	26
DBSC_ERR_WAIT_FAILED	An error occurred while waiting for the server to shutdown.	27
DBSC_ERR_SHUT-DOWN_NOT_CALLED	WaitForServerShutdown method was called without first calling the ShutdownServer method.	28
DBSC_ERR_NO_SYNC_ACK	A synchronization request was sent to the server but no acknowledgement was received; There is no way to indicate that the server received the request. hdl1 is the handle for the sync request that was sent. If the server received the request, this handle can be used to identify events for the synchronization retrieved using the GetEvent method.	29
DBSC_ERR_ACTIVE_SYNC_NOT_CANCELLED	The server could not cancel the synchronization request because the synchronization was active.	30
DBSC_ERR_DEAD_SERVER	The dbmlsync server has encountered an error while starting up. The server is now shutting down. Use the dbmlsync -o option to log the error message to a file.	31

1.4 DBSC_EventType Enumeration

Indicates the type of event generated by a synchronization.

Syntax

Visual Basic

```
Public Enum DBSC_EventType
```

C#

```
enum DBSC_EventType
```

Members

Member name	Description	Value
DBSC_EVENTTYPE_ERROR_MSG	An error was generated by the synchronization; str1 contains the text of the error.	1
DBSC_EVENTTYPE_WARNING_MSG	A warning was generated by the synchronization; str1 contains the text of the warning.	2
DBSC_EVENTTYPE_INFO_MSG	An information message was generated by the synchronization; str1 contains the text of the message.	3
DBSC_EVENTTYPE_PROGRESS_INDEX	Provides information for updating a progress bar; val1 contains the new progress value. This value can range from 0 to 1000, where 0 indicates 0% done and 1000 indicates 100% done.	4
DBSC_EVENTTYPE_PROGRESS_TEXT	The text associated with the progress bar has been updated and str1 contains the new value.	5
DBSC_EVENTTYPE_TITLE	The title for the synchronization window/control has changed and str1 contains the new title.	6
DBSC_EVENTTYPE_SYNC_START	The synchronization has begun; there is no additional information associated with this event.	7
DBSC_EVENTTYPE_SYNC_DONE	The synchronization is complete; val1 contains the exit code from the synchronization. A 0 value indicates success. A non-zero value indicates that the synchronization failed.	8
DBSC_EVENTTYPE_ML_CONNECT	A connection to the MobiLink Server was established; str1 indicates the communication protocol being used and str2 contains the network protocol options used.	10
DBSC_EVENTTYPE_UPLOAD_COMMITTED	The MobiLink server confirmed that it successfully committed the upload to the consolidated database.	11

Member name	Description	Value
DBSC_EVENTTYPE_DOWNLOAD_COMMITTED	The download has been successfully committed in the remote database. val1 contains the number of insert/update operations committed. val2 contains the number of delete operations committed	12
DBSC_EVENTTYPE_UPLOAD_START	The remote has begun to send an upload to the server.	13
DBSC_EVENTTYPE_UPLOAD_SENT	The remote has completed sending an upload segment to the server. For incremental and transactional uploads a separate event is generated each time an upload segment is sent. val1 contains the number of insert operations sent. val2 contains the number of update operations sent. val3 contains the number of delete operations sent.	14
DBSC_EVENTTYPE_DOWNLOAD_START	The remote has begun processing the download received from the server.	15

Related Information

[DBSC_Event Structure \[page 33\]](#)

[GetEvent\(out DBSC_Event, UInt32\) Method \[page 13\]](#)

1.5 DBSC_GetEventRet Enumeration

Indicates the result of an attempt to retrieve an event.

Syntax

Visual Basic

```
Public Enum DBSC_GetEventRet
```

C#

```
enum DBSC_GetEventRet
```

Members

Member name	Description	Value
DBSC_GETEVENT_OK	Indicates that an event was successfully retrieved.	1
DBSC_GETEVENT_TIMED_OUT	Indicates that the timeout expired without any event being available to return.	2
DBSC_GETEVENT_FAILED	Indicates that no event was returned because of an error condition.	3

Related Information

[GetEvent\(out DBSC_Event, UInt32\) Method \[page 13\]](#)

1.6 DBSC_ShutdownType Enumeration

Indicates how urgently the server should be shut down.

Syntax

Visual Basic

```
Public Enum DBSC_ShutdownType
```

C#

```
enum DBSC_ShutdownType
```

Members

Member name	Description	Value
DBSC_SHUT-DOWN_ON_EMPTY_QUEUE	Indicates that the server should complete any outstanding synchronization requests and then shutdown. Once the server receives the shutdown request, it does not accept any more synchronization requests.	1

Member name	Description	Value
DBSC_SHUTDOWN_CLEANLY	<p>Indicates that the server should shut-down cleanly, as quickly as possible.</p> <p>If there are outstanding synchronization requests, they are not performed and if there is a running synchronization it may be interrupted.</p>	2

Related Information

[ShutdownServer\(DBSC_ShutdownType\) Method \[page 20\]](#)

1.7 DBSC_StartType Enumeration

Indicates the action taken during a dbmsync server startup attempt.

☰ Syntax

Visual Basic

```
Public Enum DBSC_StartType
```

C#

```
enum DBSC_StartType
```

Members

Member name	Description	Value
DBSC_SS_STARTED	Indicates that a new dbmsync server was started.	1
DBSC_SS_ALREADY_RUNNING	Indicates that an existing dbmsync server was found, so no new server was started.	2

Related Information

[StartServer\(Int32, String, UInt32, out DBSC_StartType\) Method \[page 21\]](#)

1.8 DBSC_ErrorInfo Structure

Contains information about the failure of a previous method call.

Syntax

Visual Basic

```
Structure DBSC_ErrorInfo
```

C#

```
typedef struct DBSC_ErrorInfo
```

Members

All members of DBSC_ErrorInfo, including inherited members.

Variables

Modifier and Type	Variable	Description
public DBSC_ErrorType	type	Contains a value that indicates the reason for failure. Supported values are listed in the DBSC_ErrorType enumeration.
public String	str1	Contains additional information about the failure. The meaning of this information depends on the value of the type variable.
public String	str2	Contains additional information about the failure. The meaning of this information depends on the value of the type variable.
public Int32	val1	Contains additional information about the failure. The meaning of this information depends on the value of the type variable.
public Int32	val2	Contains additional information about the failure. The meaning of this information depends on the value of the type variable.

Modifier and Type	Variable	Description
public UInt32	hdl1	Contains additional information about the failure. The meaning of this information depends on the value of the type variable.

Remarks

str1, str2, val1, val2 and hdl1 contain additional information about the failure, and their meanings depend on the error type. The following error types use fields in this structure to store additional information:

- DBSC_ERR_CONNECTION_REJECTED
- DBSC_ERR_SERVER_SIDE_ERROR
- DBSC_ERR_NO_SYNC_ACK

Related Information

[DBSC_ErrorType Enumeration \[page 25\]](#)

1.9 DBSC_Event Structure

Contains information about an event generated by a synchronization.

☰ Syntax

Visual Basic

```
Structure DBSC_Event
```

C#

```
typedef struct DBSC_Event
```

Members

All members of DBSC_Event, including inherited members.

Variables

Modifier and Type	Variable	Description
public UInt32	hdl	Indicates the synchronization that generated the event. This value matches the value returned by the Sync method.
public DBSC_EventType	type	Indicates the type of event being reported.
public String	str1	Contains additional information about the event. The meaning of this information depends on the value of the type variable.
public String	str2	Contains additional information about the event. The meaning of this information depends on the value of the type variable.
public Int32	val1	Contains additional information about the event. The meaning of this information depends on the value of the type variable.
public Int32	val2	Contains additional information about the event. The meaning of this information depends on the value of the type variable.
public Int32	val3	Contains additional information about the event. The meaning of this information depends on the value of the type variable.

Related Information



[DBSC_EventType Enumeration \[page 27\]](#)

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.

About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
 - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
 - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering a SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.

The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

© 2022 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see <https://www.sap.com/about/legal/trademark.html> for additional trademark information and notices.