The Dbmlsync API

A whitepaper from Sybase iAnywhere

Author: Joshua Savill, Product Manager

Date: October 30th, 2008

This whitepaper was written in the context of SQL Anywhere 11.

SYBASE iAnywhere

CONTENTS

FIGURES

INTRODUCTION

In SQL Anywhere 11, the Dbmlsync API provides a programming interface that allows applications written in C++ or .NET development languages to launch MobiLink synchronization seamlessly. Feedback information generated by the synchronization is provided back to the application.

A .NET example of the Dbmlsync API can be found on the Sybase iAnywhere website here: http://www.sybase.com/detail?id=1060464

DBMSLYNC API ARCHITECTURE

Synchronization typically begins when the MobiLink client opens a connection with the MobiLink server. The MobiLink client uploads data changes made to the remote database since the last complete synchronization. The MobiLink server applies the data changes to the consolidated database, and then generates a download of all changes in the consolidated database since the last complete synchronization. The MobiLink client applies the changes to the remote database. Figure 1 shows the typical MobiLink architecture.

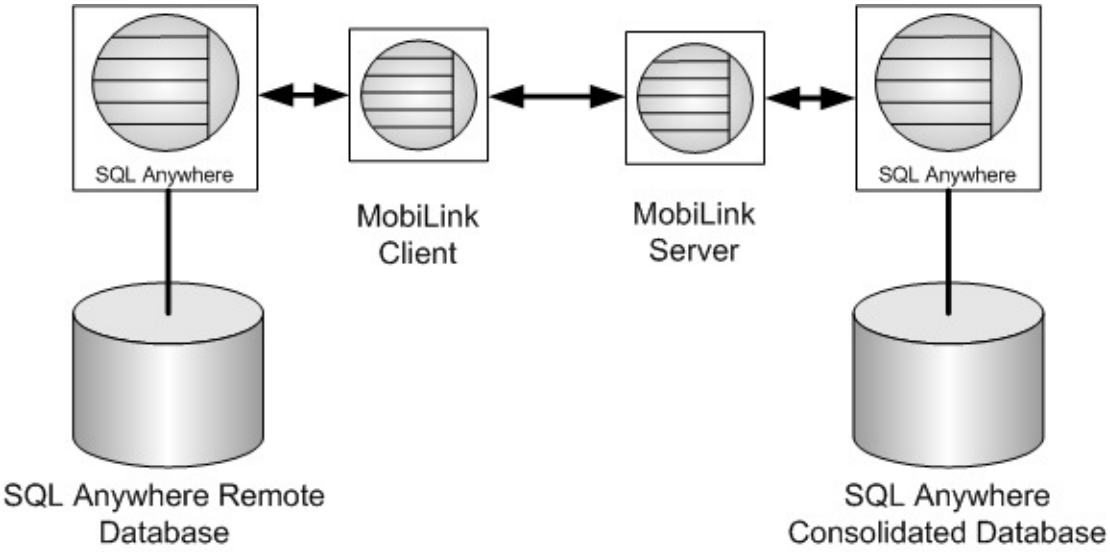

Figure 1 – MobiLink Synchronization

When using the Dbmlsync API, synchronization is architecturally the same as for a typical MobiLink synchronization. The MobiLink client still communicates with the MobiLink server to perform the synchronization. The difference is the way the application communicates with the MobiLink client to initiate communication with the MobiLink server. When using the Dbmlsync API, the MobiLink client is dbmlsync.

There are two versions of the Dbmlsync API: C++ and .NET.

The C++ version is implemented in the DLL *dbmlsynccli11.dll*. Applications that use the C++ version need to include the header *dbmlsynccli.hpp* in the C++ code and link against the import library *dbmlsynccli11.lib*. The *dbmlsynccli11.dll* must be deployed with the application.

The .NET version is implemented in the *iAnywhere.MobiLink.Client.dll* assembly. Applications using the .NET version must include a reference to the assembly in the .NET project. The *iAnywhere.MobiLink.Client.dll* assembly must be deployed with the application.

To use the Dbmlsync API, the client application instantiates and calls methods provided in the DbmlsyncClient class. The DbmlsyncClient class communicates with the dbmlsync server, using TCP/IP, to initiate a synchronization. The dbmlsync server performs the synchronization by connecting to the MobiLink server and SQL Anywhere remote database. The dbmlsync server is actually the dbmlsync client running in a listening mode waiting for synchronization requests. The dbmlsync server and application must be running on the same machine. Figure 2 shows the architecture of the Dbmlsync API and the dbmlsync server.

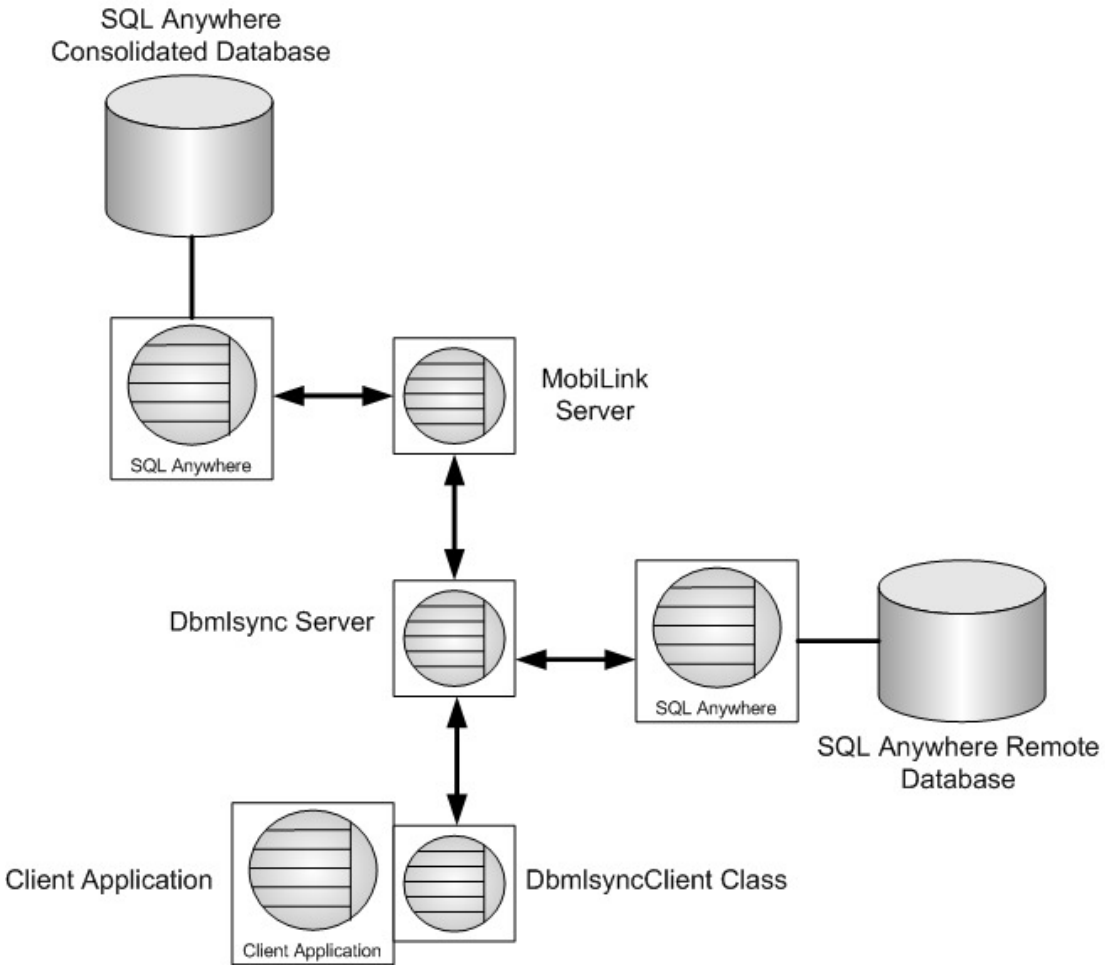

Figure 2 – Dbmlsync API and Dbmlsync Server

During a synchronization, the dbmlsync server provides a series of events that contain information and feedback about the progress of the synchronization. These events are sent from the dbmlsync server to the DbmlsyncClient class and queued. The GetEvent method, which is part of the DbmlsyncClient class, is called by the client application to retrieve the next queued event.

In this architecture, the dbmlsync server can only be connected to and synchronize a single remote database. Client applications have the ability to share a dbmlsync server, however only one synchronization can be performed on a remote database at a time. If the dbmlsync server receives a synchronization request while performing a synchronization, the request is queued and executed by the dbmlsync server when the first synchronization is complete.

DBMLSYNC API DOCUMENTATION

The Dbmlsync API programming interface is fully explained in the SQL Anywhere 11 documentation.

DBMLSYNC API FOR C++

The Dbmlsync API methods for the C++ implementation of the DbmlsyncClient class are documented in the following section of the SQL Anywhere 11 documentation:

MobiLink - Client Administration »
SQL Anywhere Clients for MobiLink »
Dbmlsync API »
Dbmlsync API for C++

Online:
http://dcx.sybase.com/index.php#http%3A%2F%2Fdcx.sybase.com%2F1100en%2Fmlclient_en11%2Fmc-dbmlsyncserv-s-4174851.html

DBMLSYNC API FOR .NET

The Dbmlsync API methods for the .NET implementation of the DbmlsyncClient class are documented in the following section of the SQL Anywhere 11 documentation:

MobiLink - Client Administration »
SQL Anywhere Clients for MobiLink »
Dbmlsync API »
Dbmlsync API for .NET

Online:
http://dcx.sybase.com/index.php#http%3A%2F%2Fdcx.sybase.com%2F1100en%2Fmlclient_en11%2Fmc-dbmlsyncserv-s-4311724.html

USING THE DBSC_EVENT STRUCTURE

While the dbmlsync server is executing a synchronization it generates a series of events that contain information and feedback about the progress of the synchronization. Each event is sent from the dbmlsync server to the DbmlsyncClient class and stored in a DBSC_Event structure. The DBSC_Event structures are queued during the synchronization. A call to the GetEvent method retrieves the next DBSC_Event structure that is queued. If no DBSC_Event structure can be retrieved, the GetEvent method waits until the next one is available, or times out based on the specified expired time.

The types of events generated during a synchronization can be controlled by a series of properties that are applied when the GetEvent method is called. The SetProperty method can enable and disable certain types of events during the synchronization. Disabling events that are not required can improve performance of the overall synchronization.

The DBSC_Event Structure is documented in the following sections of the SQL Anywhere 11 documentation:

MobiLink - Client Administration »
SQL Anywhere Clients for MobiLink »
Dbmlsync API »
Dbmlsync API for C++ »
DBSC_Event structure

Online:
http://dcx.sybase.com/index.php#http%3A%2F%2Fdcx.sybase.com%2F1100en%2Fmlclient_en11%2Fmc-dbmlsyncserv-s-5226880.html

MobiLink - Client Administration »
SQL Anywhere Clients for MobiLink »
Dbmlsync API »
Dbmlsync API for .NET »
DBSC_Event structure

Online:
http://dcx.sybase.com/index.php#http%3A%2F%2Fdcx.sybase.com%2F1100en%2Fmlclient_en11%2Fmc-dbmlsyncserv-s-5226880net.html

USING THE DBMLSYNC API

The DbmlsyncClient class needs to be instantiated before using the Dbmlsync API.

For C++, include the header *dbmlsynccli.hpp* found in the *%SQLANY11%|SDK|Include* folder and link against *dbmlsynccli.lib* found in the *%SQLANY11%|SDK|Lib|X86* folder. Deploy the application with the *dbmlsynccli.11.dll* found in the *%SQLANY11%|bin32* folder

For .NET, reference must be made to the iAnywhere.MobiLink.Client component/DLL. Typically this assembly is already integrated into the .NET components list, but if not there, it can be found in the *%SQLANY11%|Assembly|V2* directory.

Figure 3 shows the sequence flow when using the Dbmlsync API for synchronization.

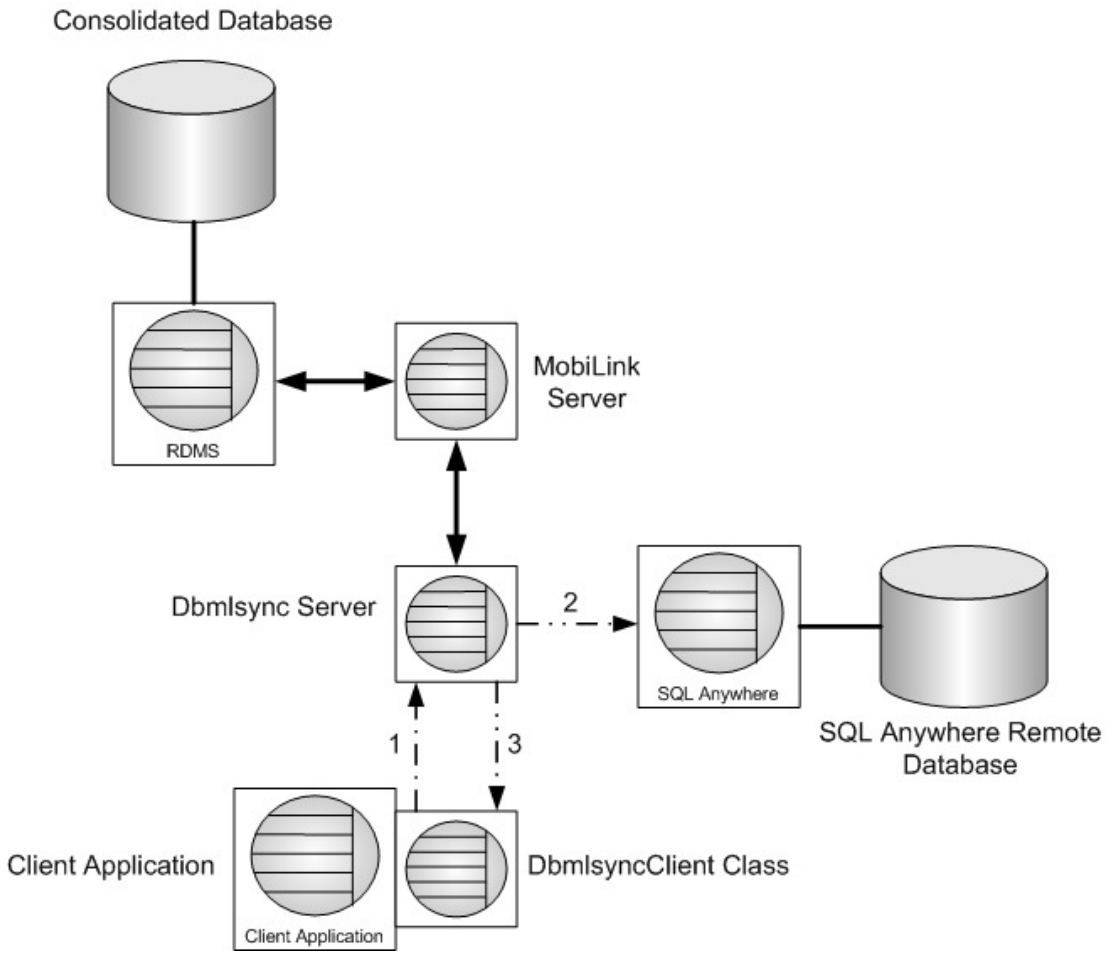

Figure 3 – Sequence Flow for Dbmlsync API

The sequence of flow for the Dbmlsync API is broken into the 3 phases. During phase 1, the client application instantiates the DbmlsyncClient class, starts the dbmlsync server, and establishes a connection with the dbmlsync server.

In phase 2, the client application calls the Sync method to request that the dbmlsync server perform a synchronization. The dbmlsync server will establish a connection with the database at this time. All other synchronization requests are queued.

While the synchronization is being performed, phase 3 shows the dbmlsync server generating a series of events that contain information and feedback about the progress. Each event is sent from the dbmlsync server to the DbmlsyncClient class and stored in a DBSC_Event structure. The DBSC_Event structures are queued. The GetEvent method is used to retrieve queued DBSC_Event structures.

STARTING THE DBMLSYNC SERVER

Here is a .NET example showing how to start the dbmlsync server:

```
using System;
using System.Collections.Generic;
using System.Text;
using iAnywhere.MobiLink.Client;

namespace StartServer {
```

```
    class Program {
        static void Main(string[] args) {
            DbmlsyncClient client = DbmlsyncClient.InstantiateClient();
            DBSC_StartType starttype;

            client.Init();
            client.StartServer(2700, "-c \"DSN=api_rem\"", 3000, out
starttype);
            client.Fini();
        }
    }
}
```

SYNCHRONIZING WITH THE DBMLSYNC SERVER

Here is a .NET example showing how to request a synchronization with the dbmlsync server. The dbmlsync server is already running with StartServer .NET execution above:

```
using System;
using System.Collections.Generic;
using System.Text;
using iAnywhere.MobiLink.Client;

namespace Sync {
    class Program {
        static void Main(string[] args) {
            DbmlsyncClient client = DbmlsyncClient.InstantiateClient();
            DBSC_Event evt = new DBSC_Event();

            client.Init();
            client.Connect("localhost", 2700, "dba", "sql");
            client.Sync("profile1", "Verbosity=HIGH");

            //Loop until DBSC_EVENTTYPE_SYNC_DONE
            while (client.GetEvent(ref evt, 3000) ==
DBSC_GetEventRet.DBSC_GETEVENT_OK) {
                    //do stuff
                    if (evt.type ==
      DBSC_EventType.DBSC_EVENTTYPE_SYNC_DONE) {
                        break;
                    }
             }
             client.Disconnect();
             client.Fini();
        }
    }
}
```

SHUTTING DOWN THE DBMLSYNC SERVER

Here is a .NET example showing how to shut down the dbmlsync server.

```
using System;
using System.Collections.Generic;
using System.Text;
```

```
using iAnywhere.MobiLink.Client;

namespace StopServer {
    class Program {
        static void Main(string[] args) {
            DbmlsyncClient client = DbmlsyncClient.InstantiateClient();

            client.Init();
            client.Connect("localhost", 2700, "dba", "sql");

client.ShutdownServer(DBSC_ShutdownType.DBSC_SHUTDOWN_ON_EMPTY_QUEUE);
            client.WaitForServerShutdown(5000);
            client.Fini();
        }
    }
}
```

SUMMARY

This document outlines how the Dbmlsync API works and provides documentation links to explain the API. Also provided are .NET examples to show how to start the dbmlsync server, initiate synchronization, and shutdown the dbmlsync server.

Copyright