



# The Dbmlsync API

Author: Joshua Savill, Product Manager

Date: October 30th, 2008

このホワイトペーパーは、SQLAnywhere 11 を対象に書かれました。

## もくじ

はじめに	3
DBMSLYNC API のアーキテクチャ	3
DBMSLYNC API のマニュアル	4
C++ 用 DBMSLYNC API	5
.NET 用 DBMSLYNC API	5
DBMSLYNC API の使用	6
DBMSLYNC サーバの起動	7
DBMSLYNC サーバとの同期	7
DBMSLYNC サーバのシャットダウン	8
まとめ	8

## 図

図 1 - Mobile Link 同期	3
図 2 - Dbmsync API と dbmsync サーバ	4
図 3 - Dbmsync API のシーケンス・フロー	6

## はじめに

SQL Anywhere 11 では、C++ または .NET 開発言語で記述されたアプリケーションが Mobile Link 同期をシームレスに起動できるようにするプログラミング・インタフェースが Dbmsync API により提供されます。同期によって生成されたフィードバック情報がアプリケーションに返されます。

Dbmsync API の .NET サンプルは、以下よりダウンロードしてください。

<ftp://ftp.iAnywhere.jp/public/tech/DBMSyncAPI.zip>

## DBMSYNC API の アーキテクチャ

通常、Mobile Link クライアントが Mobile Link サーバとの接続を開くときに同期が開始されます。Mobile Link クライアントは、最後に完了した同期以降にリモート・データベースに加えられたデータ変更をアップロードします。Mobile Link サーバは、データ変更を統合データベースに適用してから、最後に完了した同期以降に統合データベースに加えられたすべての変更のダウンロードを生成します。Mobile Link クライアントは、リモート・データベースに変更を適用します。図 1 は、一般的な Mobile Link のアーキテクチャを示しています。

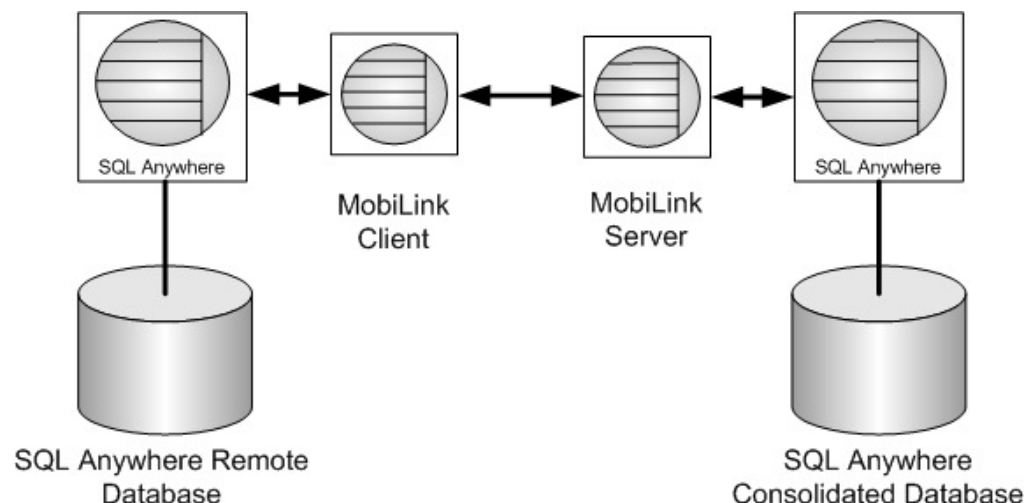


図 1 - Mobile Link 同期

Dbmsync API を使用する場合、同期のアーキテクチャは一般的な Mobile Link 同期と同じです。Mobile Link クライアントが Mobile Link サーバとの通信を介して同期を実行します。ただし、アプリケーションが Mobile Link クライアントとの通信を介して Mobile Link サーバとの通信を開始する方法は異なります。Dbmsync API を使用する場合、Mobile Link クライアントが dbmsync になります。

Dbmsync API には、C++ と .NET の 2 つのバージョンがあります。

C++ バージョンは、DLL dbmsynccli11.dll に実装されます。C++ バージョンを使用するアプリケーションは、C++ コードにヘッダ dbmsynccli.hpp をインクルードし、インポート・ライブラリ dbmsynccli11.lib とリンクする必要があります。また、dbmsynccli11.dll をアプリケーションとともに配備する必要があります。

.NET バージョンは、iAnywhere.MobiLink.Client.dll アセンブリに実装されます。NET バージョンを使用するアプリケーションは、アセンブリへの参照を .NET プロジェクトにインクルードする必要があります。また、iAnywhere.MobiLink.Client.dll アセンブリをアプリケーション

ンとともに配備する必要があります。

Dbmsync API を使用するには、クライアント・アプリケーションは、DbmsyncClient クラスで指定されているメソッドをインスタンス化して呼び出します。DbmsyncClient クラスは、TCP/IP を使用して dbmsync サーバと通信し、同期を開始します。dbmsync サーバは、Mobile Link サーバおよび SQL Anywhere リモート・データベースに接続して同期を実行します。実際には dbmsync サーバは、受信モードで同期要求を待つ dbmsync クライアントです。dbmsync サーバとアプリケーションは、同一マシン上で稼働していなければなりません。図 2 は、Dbmsync API と dbmsync サーバのアーキテクチャを示しています。

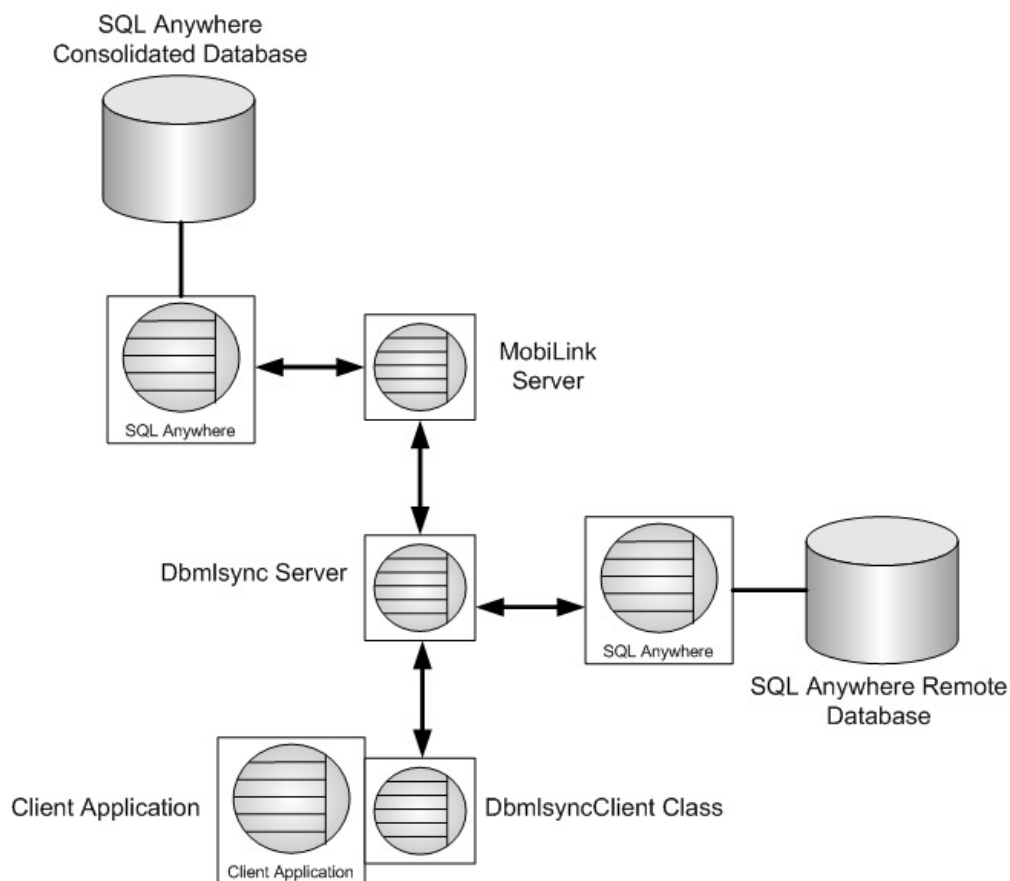


図 2 - Dbmsync API と dbmsync サーバ

同期中、dbmsync サーバが、同期の進捗状況に関する情報とフィードバックを含む一連のイベントを提供します。これらのイベントは、dbmsync サーバから DbmsyncClient クラスに送信されてキューに入ります。DbmsyncClient クラスの一部である GetEvent メソッドがクライアント・アプリケーションによって呼び出され、キュー内の次のイベントを取り出します。このアーキテクチャでは、dbmsync サーバが接続して同期できるのは、1 つのリモート・データベースだけです。クライアント・アプリケーションは、dbmsync サーバを共有する機能を持っていますが、リモート・データベースで一度に実行できる同期は 1 つだけです。dbmsync サーバが同期の実行中に同期要求を受信した場合、その要求はキューに入り、実行中の同期が完了したときに dbmsync サーバによって実行されます。

## DBMSYNC API のマニュアル

Dbmsync API プログラミング・インタフェースの詳細については、SQL Anywhere 11 のマニュアルで説明されています。

## C++ 用 DBMSYNC API

DbmsyncClient クラスの C++ 実装用 Dbmsync API メソッドについては、SQL Anywhere 11 のマニュアルの以下の項で説明されています。

Mobile Link - クライアント管理 >>  
Mobile Link 用 SQL Anywhere クライアント >>  
dbmsync API >>  
**C++ 用 dbmsync API**

オンライン・マニュアル:

[http://dcx.sybase.com/index.html#1101/ja/mlclient\\_ja11/mc-dbmsyncserv-s-4174851.html](http://dcx.sybase.com/index.html#1101/ja/mlclient_ja11/mc-dbmsyncserv-s-4174851.html)

## .NET 用 DBMSYNC API

DbmsyncClient クラスの .NET 実装用 Dbmsync API メソッドについては、SQL Anywhere 11 のマニュアルの以下の項で説明されています。

Mobile Link - クライアント管理 >>  
Mobile Link 用 SQL Anywhere クライアント >>  
dbmsync API >>  
**.NET 用 dbmsync API**

オンライン・マニュアル:

[http://dcx.sybase.com/index.html#1101/ja/mlclient\\_ja11/mc-dbmsyncserv-s-4311724.html](http://dcx.sybase.com/index.html#1101/ja/mlclient_ja11/mc-dbmsyncserv-s-4311724.html)

## DBSC.EVENT 構造体の使用

同期の実行中、dbmsync サーバは、同期の進捗状況に関する情報とフィードバックを含む一連のイベントを生成します。各イベントは、dbmsync サーバから DbmsyncClient クラスに送信され、DBSC\_Event 構造体に格納されます。同期中、DBSC\_Event 構造体はキューに入ります。GetEvent メソッドの呼び出しにより、キュー内の次の DBSC\_Event 構造体を取り出されます。DBSC\_Event 構造体を取り出せない場合、GetEvent メソッドは、次の DBSC\_Event 構造体を取り出し可能になるまで待つか、指定された時間でタイムアウトになります。

同期中に生成されるイベントの種類は、GetEvent メソッドの呼び出し時に適用される一連のプロパティによって制御できます。SetProperty メソッドにより、同期中のイベントの種類を有効化したり無効化したりすることができます。不要なイベントを無効化することで、同期全体のパフォーマンスを向上できます。

DBSC\_Event 構造体については、SQL Anywhere 11 のマニュアルの以下の項で説明されています。

Mobile Link - クライアント管理 >>  
Mobile Link 用 SQL Anywhere クライアント >>  
dbmsync API >>  
C++ 用 dbmsync API >>  
**DBSC\_Event 構造体**

オンライン・マニュアル:

[http://dcx.sybase.com/index.php#1100en/mlclient\\_en11/mc-dbmsyncserv-s-5226880.html](http://dcx.sybase.com/index.php#1100en/mlclient_en11/mc-dbmsyncserv-s-5226880.html)

Mobile Link - クライアント管理 >>  
Mobile Link 用 SQL Anywhere クライアント >>  
dbmsync API >>  
.NET 用 dbmsync API >>  
**DBSC\_Event 構造体**

オンライン・マニュアル:

[http://dcx.sybase.com/index.html#1101/ja/mlclient\\_ja11/mc-dbmsyncserv-s-5226880.html](http://dcx.sybase.com/index.html#1101/ja/mlclient_ja11/mc-dbmsyncserv-s-5226880.html)

## DBMSYNC API の使用

Dbmsync API を使用する前に、DbmsyncClient クラスをインスタンス化する必要があります。

C++ の場合、%SQLANY11%\SDK\Include フォルダ内のヘッダ dbmsynccli.hpp をインクルードし、%SQLANY11%\SDK\Lib\X86 フォルダ内の dbmsynccli.lib にリンクします。また、アプリケーションとともに %SQLANY11%\bin32 フォルダ内の dbmsynccli.11.dll を配備します。

.NET の場合、iAnywhere.MobiLink.Client コンポーネント/DLL への参照を作成する必要があります。通常、このアセンブリはすでに .NET コンポーネント・リストに統合されていますが、リストにない場合は、%SQLANY11%\Assembly\%V2 ディレクトリに配置されています。

図 3 は、Dbmsync API を使用して同期を行う場合のシーケンス・フローです。

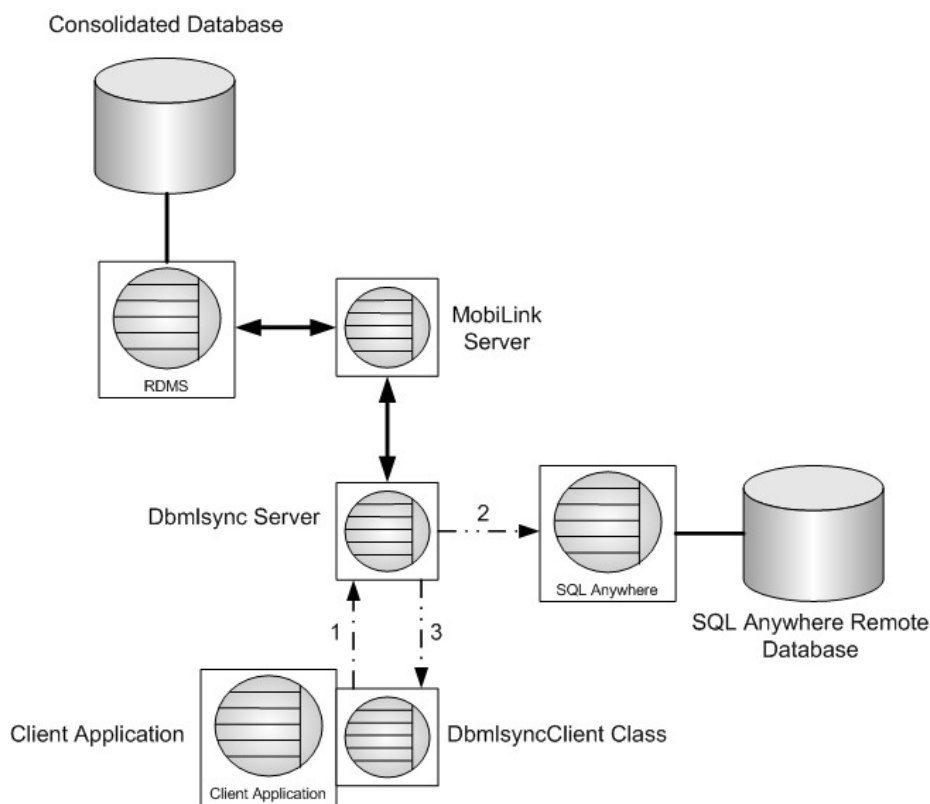


図 3 - Dbmsync API のシーケンス・フロー

Dbmsync API のシーケンス・フローは、3 つのフェーズに分けられます。フェーズ 1 では、クライアント・アプリケーションが DbmsyncClient クラスをインスタンス化し、dbmsync サーバを起動し、dbmsync サーバとの接続を確立します。

フェーズ 2 では、クライアント・アプリケーションが Sync メソッドを呼び出し、dbmsync サーバによる同期の実行を要求します。dbmsync サーバは、この時点でデータベースとの接続を確立します。その他の同期要求はすべてキューに入ります。

同期の実行中がフェーズ 3 になります。フェーズ 3 では dbmsync サーバが、同期の進捗状況に関する情報とフィードバックを含む一連のイベントを生成します。各イベントは、dbmsync サーバから DbmsyncClient クラスに送信され、DBSC\_Event 構造体に格納されます。DBSC\_Event 構造体はキューに入ります。GetEvent メソッドを使用して、キュー内の DBSC\_Event 構造体を取り出されます。

## DBMSYNC サーバの起動

以下の .NET サンプルは、dbmsync サーバの起動方法を示しています。

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

namespace StartServer {
    class Program {
        static void Main(string[] args) {
            DbmsyncClient client = DbmsyncClient.InstantiateClient();
            DBSC_StartType starttype;

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

## DBMSYNC サーバとの同期

以下の .NET サンプルは、dbmsync サーバとの同期の要求方法を示しています。dbmsync サーバは、上記の StartServer .NET の実行によりすでに稼働中であるとします。

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

namespace Sync {
    class Program {
        static void Main(string[] args) {
            DbmsyncClient client = DbmsyncClient.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();
    }
}
}
```

## DBMLSYNC サーバのシャットダウン

以下の .NET サンプルは、dbmlsync サーバのシャットダウン方法を示しています。

```
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();
        }
    }
}
```

## まとめ

このマニュアルでは、Dbmlsync API の動作について説明し、API に関するマニュアルへのリンクを提供しました。また、dbmlsync サーバの起動方法、同期の開始方法、および dbmlsync サーバのシャットダウン方法を示す .NET サンプルもご紹介しました。



## 法的注意

---

Copyright (C) 2008 iAnywhere Solutions, Inc. All rights reserved.

iAnywhere Solutions、iAnywhere Solutions（ロゴ）は、iAnywhere Solutions, Inc.とその系列会社の商標です。その他の商標はすべて各社に帰属します。

本書に記載された情報、助言、推奨、ソフトウェア、文書、データ、サービス、ロゴ、商標、図版、テキスト、写真、およびその他の資料（これらすべてを"資料"と総称する）は、iAnywhere Solutions, Inc.とその提供元に帰属し、著作権や商標の法律および国際条約によって保護されています。また、これらの資料はいずれも、iAnywhere Solutionsとその提供元の知的所有権の対象となるものであり、iAnywhere Solutionsとその提供元がこれらの権利のすべてを保有するものとします。

資料のいかなる部分も、iAnywhere Solutionの知的所有権のライセンスを付与したり、既存のライセンス契約に修正を加えることを認めるものではないものとします。

資料は無保証で提供されるものであり、いかなる保証も行われません。iAnywhere Solutionsは、資料に関するすべての陳述と保証を明示的に拒否します。これには、商業性、特定の目的への整合性、非侵害性の黙示的な保証を無制限に含みます。

iAnywhere Solutionsは、資料自体の、または資料が依拠していると思われる内容、結果、正確性、適時性、完全性に関して、いかなる理由であろうと保証や陳述を行いません。iAnywhere Solutionsは、資料が途切れていないこと、誤りがないこと、いかなる欠陥も修正されていることに関して保証や陳述を行いません。ここでは、「iAnywhere Solutions」とは、iAnywhere Solutions, Inc.またはSybase, Inc.とその部門、子会社、継承者、および親会社と、その従業員、パートナー、社長、代理人、および代表者と、さらに資料を提供した第三者の情報元や提供者を表します。