

SQL Anywhere - Mobile Link  
文書バージョン: 17 - 2016-05-11

## Mobile Link - Dbmlsync C++ API リファレンス

# 目次

<b>1</b>	<b>Dbmsync C++ API リファレンス</b> .....	<b>3</b>
1.1	DbmsyncClient クラス.....	4
	CancelSync メソッド.....	7
	Connect(const char *, unsigned, const char *, const char *) メソッド.....	9
	Disconnect(void) メソッド.....	11
	Fini(void) メソッド.....	11
	FreeEventInfo(DBSC_Event *) メソッド.....	12
	GetErrorInfo(void) メソッド.....	13
	GetEvent(DBSC_Event **, unsigned) メソッド.....	13
	GetProperty(const char *, char *) メソッド.....	15
	Init(void) メソッド.....	15
	InstantiateClient(void) メソッド.....	16
	Ping(unsigned) メソッド.....	17
	SetProperty(const char *, const char *) メソッド.....	17
	ShutdownServer(DBSC_ShutdownType) メソッド.....	19
	StartServer(unsigned, const char *, unsigned, DBSC_StartType *) メソッド.....	20
	Sync(const char *, const char *) メソッド.....	21
	WaitForServerShutdown(unsigned) メソッド.....	22
1.2	DBSC_CancelRet 列挙.....	23
1.3	DBSC_ErrorType 列挙.....	24
1.4	DBSC_EventType 列挙.....	26
1.5	DBSC_GetEventRet 列挙.....	28
1.6	DBSC_ShutdownType 列挙.....	28
1.7	DBSC_StartType 列挙.....	29
1.8	DBSC_ErrorInfo 構造体.....	30
1.9	DBSC_Event 構造体.....	31
<b>2</b>	<b>このマニュアルの印刷、再生、および再配布</b> .....	<b>33</b>

# 1 Dbmlsync C++ API リファレンス

Dbmlsync C++ API は、C++ で記述された Mobile Link クライアントアプリケーションが同期を起動し、要求した同期の進行状況に関するフィードバックを受け取れるようにするプログラミングインタフェースです。

## ヘッダファイル

```
dbmlsynccli.hpp
```

### i 注記

主な **SQL Anywhere** マニュアルをお探しですか。マニュアルをローカルにインストールした場合は、Windows のスタートメニューを使用してアクセスするか (Microsoft Windows)、C:\Program Files\SQL Anywhere 17\Documentation にナビゲートします。

また、DocCommentXchange の Web で、主な SQL Anywhere API リファレンスマニュアルにアクセスすることもできます。<http://dcx.sap.com>

### 例

この後に示す例は、Dbmlsync API の C++ バージョンを使用して同期を行い、出力イベントを受け取る典型的なアプリケーションを示したものです。この例では、わかりやすいようにエラー処理を省略しています。各 API 呼び出しの戻り値を確認する習慣をつけることをおすすめします。

```
#include <stdio.h>
#include "dbmlsynccli.hpp"
int main( void ) {
    DbmlsyncClient *client;
    DBSC_SyncHdl    syncHdl;
    DBSC_Event     *ev1;
    client = DbmlsyncClient::InstantiateClient();
    if( client == NULL ) return( 1 );
    client->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.
    client->SetProperty( "server path", "C:\SQLAnywhere\bin32" );
    client->StartServer( 3426,
        "-c server=remote;dbn=rem1;uid=dba;pwd=passwd -v+ -ot c:\
        %dbsync1.txt",
        5000, NULL );
    client->Connect( NULL, 3426, "dba", "sql");
    syncHdl = client->Sync( "my_sync_profile", "" );
    while( client->GetEvent( &ev1, 5000) == DBSC_GETEVENT_OK ) {
        if( ev1->hdl == syncHdl ) {
            //
            // Process events that interest you here
            //
            if( ev1->type == DBSC_EVENTTYPE_SYNC_DONE ) {
                client->FreeEventInfo( ev1 );
            }
        }
    }
}
```

```

        break;
    }
    client->FreeEventInfo( ev1 );
}
}
client->ShutdownServer( DBSC_SHUTDOWN_ON_EMPTY_QUEUE );
client->WaitForServerShutdown( 10000 );
client->Disconnect();
client->Fini();
delete client;
return( 0 );
}

```

このセクションの内容:

#### [DbmlsyncClient クラス \[4 ページ\]](#)

TCP/IP を使用して別のプロセス (dbmlsync サーバ) と通信します。このプロセスが、Mobile Link サーバとリモートデータベースに接続することによって同期を実行します。

#### [DBSC\\_CancelRet 列挙 \[23 ページ\]](#)

同期キャンセル試行の結果を示します。

#### [DBSC\\_ErrorType 列挙 \[24 ページ\]](#)

メソッド呼び出しが失敗した理由を示します。

#### [DBSC\\_EventType 列挙 \[26 ページ\]](#)

同期によって生成されたイベントのタイプを示します。

#### [DBSC\\_GetEventRet 列挙 \[28 ページ\]](#)

イベント取得の試行の結果を示します。

#### [DBSC\\_ShutdownType 列挙 \[28 ページ\]](#)

サーバを停止する緊急度を示します。

#### [DBSC\\_StartType 列挙 \[29 ページ\]](#)

dbmlsync サーバを起動しようとしているときに実行されたアクションを示します。

#### [DBSC\\_ErrorInfo 構造体 \[30 ページ\]](#)

以前のメソッド呼び出しの失敗に関する情報が含まれています。

#### [DBSC\\_Event 構造体 \[31 ページ\]](#)

同期によって生成されたイベントに関する情報が含まれています。

## 1.1 DbmlsyncClient クラス

TCP/IP を使用して別のプロセス (dbmlsync サーバ) と通信します。このプロセスが、Mobile Link サーバとリモートデータベースに接続することによって同期を実行します。

### 構文

```
public class DbmlsyncClient
```

## メンバー

DbmlsyncClient のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

### メソッド

変更子とタイプ	メソッド	説明
public virtual bool	<a href="#">CancelSync [7 ページ]</a>	同期要求をキャンセルします。
public virtual bool	<a href="#">Connect(const char *, unsigned, const char *, const char *) [9 ページ]</a>	すでにこのコンピュータで実行されている dbmlsync サーバとの接続を開きます。
public virtual bool	<a href="#">Disconnect(void) [11 ページ]</a>	Connect メソッドを使用して確立された dbmlsync サーバとの接続を切断します。
public virtual bool	<a href="#">Fini(void) [11 ページ]</a>	このクラスインスタンスによって使用されているすべてのリソースを解放します。
public virtual bool	<a href="#">FreeEventInfo(DBSC_Event *) [12 ページ]</a>	GetEvent メソッドによって返された DBSC_Event 構造体に関連付けられていたメモリを解放します。
public virtual const DBSC_ErrorInfo *	<a href="#">GetErrorInfo(void) [13 ページ]</a>	DbmlsyncClient クラスメソッドによって失敗を示すリターンコードが返された後、失敗に関する追加情報を取得します。
public virtual DBSC_GetEventRet	<a href="#">GetEvent(DBSC_Event **, unsigned) [13 ページ]</a>	クライアントが要求した同期の次のフィードバックイベントを取得します。
public virtual bool	<a href="#">GetProperty(const char *, char *) [15 ページ]</a>	プロパティの現在の値を取得します。
public virtual bool	<a href="#">Init(void) [15 ページ]</a>	DbmlsyncClient クラスインスタンスを初期化します。
public static DbmlsyncClient *	<a href="#">InstantiateClient(void) [16 ページ]</a>	同期の制御に使用できる dbmlsync クライアントクラスのインスタンスを作成します。
public virtual bool	<a href="#">Ping(unsigned) [17 ページ]</a>	dbmlsync サーバに ping 要求を送信して、サーバがアクティブで、要求に回答しているかどうかをチェックします。
public virtual bool	<a href="#">SetProperty(const char *, const char *) [17 ページ]</a>	各種のプロパティを設定して、クラスインスタンスの動作を変更します。
public virtual bool	<a href="#">ShutdownServer(DBSC_ShutdownType) [19 ページ]</a>	クライアントの接続先である dbmlsync サーバを停止します。
public virtual bool	<a href="#">StartServer(unsigned, const char *, unsigned, DBSC_StartType *) [20 ページ]</a>	指定したポートでまだ受信していない場合は、新しい dbmlsync サーバを起動します。
public virtual DBSC_SyncHdl	<a href="#">Sync(const char *, const char *) [21 ページ]</a>	同期を実行するよう、dbmlsync サーバに要求します。
public virtual bool	<a href="#">WaitForServerShutdown(unsigned) [22 ページ]</a>	サーバが停止したときタイムアウトになったときのどちらか早い方で戻ります。

## 備考

複数のクライアントで同じ dbmsync サーバを共有できます。ただし、各 dbmsync サーバが同期できるのは1つのリモートデータベースのみです。各リモートデータベースには、同期する dbmsync サーバを1つのみ設定できます。

dbmsync サーバは、一度に1つの同期を実行します。サーバは同期の実行中に同期要求を受信した場合、その要求をキューイングし、後で履行します。

同期によって生成されたステータス情報は、GetEvent メソッドを通じてクライアントアプリケーションに送り返されます。

このセクションの内容:

### [CancelSync メソッド \[7 ページ\]](#)

同期要求をキャンセルします。

### [Connect\(const char \\*, unsigned, const char \\*, const char \\*\) メソッド \[9 ページ\]](#)

すでにこのコンピュータで実行されている dbmsync サーバとの接続を開きます。

### [Disconnect\(void\) メソッド \[11 ページ\]](#)

Connect メソッドを使用して確立された dbmsync サーバとの接続を切断します。

### [Fini\(void\) メソッド \[11 ページ\]](#)

このクラスインスタンスによって使用されているすべてのリソースを解放します。

### [FreeEventInfo\(DBSC\\_Event \\*\) メソッド \[12 ページ\]](#)

GetEvent メソッドによって返された DBSC\_Event 構造体に関連付けられていたメモリを解放します。

### [GetErrorInfo\(void\) メソッド \[13 ページ\]](#)

DbmsyncClient クラスメソッドによって失敗を示すリターンコードが返された後、失敗に関する追加情報を取得します。

### [GetEvent\(DBSC\\_Event \\*\\*, unsigned\) メソッド \[13 ページ\]](#)

クライアントが要求した同期の次のフィードバックイベントを取得します。

### [GetProperty\(const char \\*, char \\*\) メソッド \[15 ページ\]](#)

プロパティの現在の値を取得します。

### [Init\(void\) メソッド \[15 ページ\]](#)

DbmsyncClient クラスインスタンスを初期化します。

### [InstantiateClient\(void\) メソッド \[16 ページ\]](#)

同期の制御に使用できる dbmsync クライアントクラスのインスタンスを作成します。

### [Ping\(unsigned\) メソッド \[17 ページ\]](#)

dbmsync サーバに ping 要求を送信して、サーバがアクティブで、要求に応答しているかどうかをチェックします。

### [SetProperty\(const char \\*, const char \\*\) メソッド \[17 ページ\]](#)

各種のプロパティを設定して、クラスインスタンスの動作を変更します。

### [ShutdownServer\(DBSC\\_ShutdownType\) メソッド \[19 ページ\]](#)

クライアントの接続先である dbmsync サーバを停止します。

### [StartServer\(unsigned, const char \\*, unsigned, DBSC\\_StartType \\*\) メソッド \[20 ページ\]](#)

指定したポートでまだ受信していない場合は、新しい dbmsync サーバを起動します。

### [Sync\(const char \\*, const char \\*\) メソッド \[21 ページ\]](#)

同期を実行するよう、dbmsync サーバに要求します。

[WaitForServerShutdown\(unsigned\) メソッド \[22 ページ\]](#)

サーバが停止したときかタイムアウトになったときのどちらか早い方で戻ります。

## 関連情報

[GetEvent\(DBSC\\_Event \\*\\*, unsigned\) メソッド \[13 ページ\]](#)

### 1.1.1 CancelSync メソッド

同期要求をキャンセルします。

#### オーバーロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual bool	<a href="#">CancelSync(DBSC_SyncHdl) [7 ページ]</a>	Sync メソッドを使用して直前に行われた同期要求を、クライアントがキャンセルできるようにします。
public virtual DBSC_CancelRet	<a href="#">CancelSync(DBSC_SyncHdl, bool) [8 ページ]</a>	Sync メソッドを使用して直前に行われた同期要求を、クライアントがキャンセルできるようにします。

このセクションの内容:

[CancelSync\(DBSC\\_SyncHdl\) メソッド \(旧式\) \[7 ページ\]](#)

Sync メソッドを使用して直前に行われた同期要求を、クライアントがキャンセルできるようにします。

[CancelSync\(DBSC\\_SyncHdl, bool\) メソッド \[8 ページ\]](#)

Sync メソッドを使用して直前に行われた同期要求を、クライアントがキャンセルできるようにします。

#### 1.1.1.1 CancelSync(DBSC\_SyncHdl) メソッド (旧式)

Sync メソッドを使用して直前に行われた同期要求を、クライアントがキャンセルできるようにします。

##### 構文

```
public virtual bool CancelSync (DBSC_SyncHdl hdl)
```

## パラメータ

hdl 同期が要求されたときに Sync メソッドによって返された同期ハンドル。

## 戻り値

同期要求が正常にキャンセルされた場合は true、正常にキャンセルされなかった場合は false を返します。false が返されたときは、GetErrorInfo メソッドを呼び出して、失敗に関する詳細な情報を取得できます。

## 備考

キャンセルできるのは、サービスが提供されるのを待っている同期要求のみです。すでに開始された同期を停止するには、CancelSync(UInt32, Boolean) メソッドを使用します。

このメソッドを使用するには、サーバへの接続が確立されている必要があります。Sync メソッドを呼び出した後、クライアントがサーバから切断されている場合は、このメソッドを使用することはできません。

## 関連情報

[GetErrorInfo\(void\) メソッド \[13 ページ\]](#)

[CancelSync\(DBSC\\_SyncHdl, bool\) メソッド \[8 ページ\]](#)

[ShutdownServer\(DBSC\\_ShutdownType\) メソッド \[19 ページ\]](#)

### 1.1.1.2 CancelSync(DBSC\_SyncHdl, bool) メソッド

Sync メソッドを使用して直前に行われた同期要求を、クライアントがキャンセルできるようにします。

#### 構文

```
public virtual DBSC_CancelRet CancelSync (
    DBSC_SyncHdl hdl,
    bool cancel_active
)
```

## パラメータ

hdl 同期が要求されたときに Sync メソッドによって返された同期ハンドル。



**cancel\_active** true に設定されている場合は、同期がすでに開始していても要求がキャンセルされます。false に設定されている場合は、同期が開始していないときにのみ要求がキャンセルされます。

## 戻り値

DBSC\_CancelRet 列挙の値。DBSC\_CANCEL\_FAILED が返されたときは、GetErrorInfo メソッドを呼び出して、失敗に関する詳細な情報を取得できます。

## 備考

このメソッドを使用するには、サーバへの接続が確立されている必要があります。Sync メソッドを呼び出した後、クライアントがサーバから切断されている場合は、このメソッドを使用することはできません。

## 関連情報

[DBSC\\_CancelRet 列挙 \[23 ページ\]](#)

[GetErrorInfo\(void\) メソッド \[13 ページ\]](#)

[ShutdownServer\(DBSC\\_ShutdownType\) メソッド \[19 ページ\]](#)

## 1.1.2 Connect(const char \*, unsigned, const char \*, const char \*) メソッド

すでにこのコンピュータで実行されている dbmsync サーバとの接続を開きます。

### 構文

```
public virtual bool Connect (  
    const char * host,  
    unsigned port,  
    const char * uid,  
    const char * pwd  
)
```

## パラメータ

**host** この値は予約されています。NULL を使用します。

**port** dbmsync サーバが受信している TCP ポート。StartServer メソッドを使用して指定した port 値と同じ port 値を使用します。

**uid** 同期されるリモートデータベースに対する DBA 権限または REMOTE DBA 権限を持つ、有効なデータベースユーザ ID。

**pwd** uid で指定されたユーザのデータベースパスワード。

## 戻り値

サーバへの接続が確立された場合は true、確立されなかった場合は false を返します。false が返されたときは、GetErrorInfo メソッドを呼び出して、失敗に関する詳細な情報を取得できます。

## 備考

データベース userid およびパスワード (*server\_userid* など) を提供する接続文字列を使用して、(コマンドラインまたは StartServer メソッドで) Dbmsync サーバを起動します。さらに、Dbmsync API の Connect メソッドは、有効なデータベース userid (*client\_userid* など) を必要とします。

*Client\_userid* は、このクライアントにデータベースを同期するための十分なパーミッションがあるかどうかを検証するためだけに使用されます。同期が実行されると、*server\_userid* が使用されます。

SQL Anywhere 12 以前では、*client\_userid* と *server\_userid* の両方で DBA または REMOTE DBA パーミッションが必要でした。

SQL Anywhere 16 以降では、*server\_userid* が同期に十分な権限を持っている必要があります。最低でも、*server\_userid* には SYS\_RUN\_REPLICATION\_ROLE システムロールが必要ですが、同期できるために他の権限が必要になることがあります。*Client\_userid* は、次のいずれかになっている必要があります。

- *server\_userid* と同じであるか、
- SYS\_AUTH\_DBA\_ROLE システムロールを持っているか、
- *server\_userid* に基づくユーザ拡張ロールを持っている (CREATE ROLE FOR USER *server\_userid*、GRANT *server\_userid* to *client\_userid* など)。

最後のオプションが、*client\_userid* に少なくとも *server\_userid* と同じ数のシステム権限があることを確認している間、*server\_userid* がロールにオブジェクトレベルの権限を付与されていず、したがって *client\_userid* にはこれらの権限がないということもあります。これらの権限が同期時に使用される場合、*client\_userid* は同期を実行するための権限を効率的に増やします。これを許容できない場合、すべての *server\_userid*s オブジェクトレベル権限がユーザ拡張ロールに付与されていることを確認します。

## 関連情報

[StartServer\(unsigned, const char \\*, unsigned, DBSC\\_StartType \\*\) メソッド \[20 ページ\]](#)

[GetErrorInfo\(void\) メソッド \[13 ページ\]](#)

## 1.1.3 Disconnect(void) メソッド

Connect メソッドを使用して確立された dbmlsync サーバとの接続を切断します。

### 構文

```
public virtual bool Disconnect (void)
```

### 戻り値

サーバへの接続が切断された場合は true、切断されなかった場合は false を返します。false が返されたときは、GetErrorInfo メソッドを呼び出して、失敗に関する詳細な情報を取得できます。

### 備考

接続を使い終わったら、必ず Disconnect を呼び出してください。

### 関連情報

[Connect\(const char \\*, unsigned, const char \\*, const char \\*\) メソッド \[9 ページ\]](#)

[GetErrorInfo\(void\) メソッド \[13 ページ\]](#)

## 1.1.4 Fini(void) メソッド

このクラスインスタンスによって使用されているすべてのリソースを解放します。

### 構文

```
public virtual bool Fini (void)
```

### 戻り値

クラスインスタンスが正常に終了された場合は true、正常に終了されなかった場合は false を返します。false が返されたときは、GetErrorInfo メソッドを呼び出して、失敗に関する詳細な情報を取得できます。

## 備考

DbmlSyncClient クラスインスタンスを削除するには、このメソッドを呼び出しておく必要があります。

### **i** 注記

Disconnect メソッドを使用して接続しているすべてのサーバを切断してから、クラスインスタンスを終了してください。

## 関連情報

[Disconnect\(void\) メソッド \[11 ページ\]](#)

[GetErrorInfo\(void\) メソッド \[13 ページ\]](#)

## 1.1.5 FreeEventInfo(DBSC\_Event \*) メソッド

GetEvent メソッドによって返された DBSC\_Event 構造体に関連付けられていたメモリを解放します。

### 構文

```
public virtual bool FreeEventInfo (DBSC_Event * event)
```

## パラメータ

**event** 解放する DBSC\_Event 構造体へのポインタ。

## 戻り値

メモリが正常に解放された場合は true、正常に解放されなかった場合は false を返します。false が返されたときは、GetErrorInfo メソッドを呼び出して、失敗に関する詳細な情報を取得できます。

## 備考

GetEvent メソッドによって返された DBSC\_Event 構造体それぞれに対して、FreeEventInfo を呼び出してください。

## 関連情報

[DBSC\\_Event 構造体 \[31 ページ\]](#)

[GetEvent\(DBSC\\_Event \\*\\*, unsigned\) メソッド \[13 ページ\]](#)

[GetErrorInfo\(void\) メソッド \[13 ページ\]](#)

### 1.1.6 GetErrorInfo(void) メソッド

DbmlsyncClient クラスメソッドによって失敗を示すリターンコードが返された後、失敗に関する追加情報を取得します。

#### 構文

```
public virtual const DBSC_ErrorInfo * GetErrorInfo (void)
```

## 戻り値

失敗に関する情報を含む DBSC\_ErrorInfo 構造体へのポインタ。この構造体の内容は、クラスメソッドが次回呼び出されたときに上書きされることがあります。

## 関連情報

[DBSC\\_ErrorType 列挙 \[24 ページ\]](#)

[DBSC\\_ErrorInfo 構造体 \[30 ページ\]](#)

### 1.1.7 GetEvent(DBSC\_Event \*\*, unsigned) メソッド

クライアントが要求した同期の次のフィードバックイベントを取得します。

#### 構文

```
public virtual DBSC_GetEventRet GetEvent (
    DBSC_Event ** event,
    unsigned timeout
)
```

## パラメータ

**event** 戻り値が DBSC\_GETEVENT\_OK である場合、event パラメータに、取得されたイベントに関する情報を含む DBSC\_Event 構造体へのポインタが入力されます。event の構造体を使い終わったら、FreeEventInfo メソッドを呼び出して、event の構造体に関連付けられていたメモリを解放してください。

**timeout** すぐに返すことができるイベントがない場合に待つ最大のミリ秒数を指定します。DBSC\_INFINITY を使用して応答を無期限に待機します。

## 戻り値

DBSC\_GetEventRet 列挙の値。DBSC\_GETEVENT\_FAILED が返されたときは、GetErrorInfo メソッドを呼び出して、失敗に関する詳細な情報を取得できます。

## 備考

フィードバックイベントには、sync から生成されたメッセージ、進行状況バーを更新するためのデータ、同期サイクル通知などの情報が含まれています。

dbmlsync サーバは、同期を実行しながら、同期の進行状況に関する情報を含む一連のイベントを生成します。これらのイベントは、サーバから DbmlsyncClient クラスに送信され、そこでキューイングされます。GetEvent メソッドを呼び出すと、キュー内で待機している次のイベントがある場合、そのイベントが返されます。

キュー内で待機しているイベントがない場合、このメソッドは、イベントが実行可能になるまで、または指定されたタイムアウトになるまで待つから、戻ります。

同期用に生成されるイベントのタイプは、プロパティを使用して制御できます。

## 関連情報

[DBSC\\_GetEventRet 列挙 \[28 ページ\]](#)

[DBSC\\_Event 構造体 \[31 ページ\]](#)

[FreeEventInfo\(DBSC\\_Event \\*\) メソッド \[12 ページ\]](#)

[SetProperty\(const char \\*, const char \\*\) メソッド \[17 ページ\]](#)

[GetErrorInfo\(void\) メソッド \[13 ページ\]](#)

## 1.1.8 GetProperty(const char \*, char \*) メソッド

プロパティの現在の値を取得します。

### 構文

```
public virtual bool GetProperty (  
    const char * name,  
    char * value  
)
```

### パラメータ

**name** 取り出すプロパティの名前。有効なプロパティ名のリストについては、SetProperty を参照してください。

**value** プロパティの値を格納する、DBSC\_MAX\_PROPERTY\_LEN バイト以上のバッファ。

### 戻り値

プロパティが正常に受信された場合は true、正常に受信されなかった場合は false を返します。false が返されたときは、GetErrorInfo メソッドを呼び出して、失敗に関する詳細な情報を取得できます。

### 関連情報

[SetProperty\(const char \\*, const char \\*\) メソッド \[17 ページ\]](#)

[GetErrorInfo\(void\) メソッド \[13 ページ\]](#)

## 1.1.9 Init(void) メソッド

DbmlsyncClient クラスインスタンスを初期化します。

### 構文

```
public virtual bool Init (void)
```

## 戻り値

クラスインスタンスが正常に初期化された場合は true、正常に初期化されなかった場合は false を返します。false が返されたときは、GetErrorInfo メソッドを呼び出して、失敗に関する詳細な情報を取得できます。

## 備考

DbmlSyncClient クラスインスタンスをインスタンス化した後、このメソッドを呼び出してください。インスタンスが正常に初期化されるまで、他の DbmlSyncClient メソッドを呼び出すことはできません。

## 関連情報

[InstantiateClient\(void\) メソッド \[16 ページ\]](#)

[GetErrorInfo\(void\) メソッド \[13 ページ\]](#)

## 1.1.10 InstantiateClient(void) メソッド

同期の制御に使用できる dbmlsync クライアントクラスのインスタンスを作成します。

### 構文

```
public static DbmlsyncClient * InstantiateClient (void)
```

## 戻り値

作成された新しいインスタンスへのポインタ。エラーが発生した場合は NULL を返します。

## 備考

このメソッドによって返されるポインタを使用して、クラス内の残りのメソッドを呼び出すことができます。ポインタで標準の削除操作を呼び出すことで、インスタンスを破棄できます。



## 1.1.11 Ping(unsigned) メソッド

dbmlsync サーバに ping 要求を送信して、サーバがアクティブで、要求に応答しているかどうかをチェックします。

### 構文

```
public virtual bool Ping (unsigned timeout)
```

### パラメータ

**timeout** サーバが ping 要求に応答するのを待つ最大のミリ秒数。DBSC\_INFINITY を使用して応答を無期限に待機します。

### 戻り値

ping 要求に対する応答をサーバから受信した場合は true、受信しなかった場合は false を返します。false が返されたときは、GetErrorInfo メソッドを呼び出して、失敗に関する詳細な情報を取得できます。

### 備考

サーバに接続してから、このメソッドを呼び出してください。

### 関連情報

[GetErrorInfo\(void\) メソッド \[13 ページ\]](#)

## 1.1.12 SetProperty(const char \*, const char \*) メソッド

各種のプロパティを設定して、クラスインスタンスの動作を変更します。

### 構文

```
public virtual bool SetProperty (  
    const char * name,  
    const char * value  
)
```

## パラメータ

**name** 設定するプロパティの名前。有効なプロパティ名のリストについては、表を参照してください。

**value** プロパティに設定する値。指定する文字列は、DBCS\_MAX\_PROPERTY\_LEN バイト未満にしてください。

## 戻り値

プロパティが正常に設定された場合は true、正常に設定されなかった場合は false を返します。false が返されたときは、GetErrorInfo メソッドを呼び出して、失敗に関する詳細な情報を取得できます。

## 備考

プロパティ値への変更は、変更後に行われた同期要求にのみ反映されます。

**server path** プロパティを設定して、StartServer メソッドが呼び出されたときにクライアントが dbmsync.exe を起動するディレクトリを指定できます。このプロパティが設定されていない場合、PATH 環境変数を使用して dbmsync.exe が検索されます。コンピュータに複数のバージョンの SQL Anywhere がインストールされている場合は、**server path** プロパティを使用して dbmsync.exe のロケーションを指定することをお奨めします。これは、PATH 環境変数によって、インストールされた別のバージョンの SQL Anywhere の dbmsync 実行プログラムが検出される可能性があるためです。例:

```
ret = cli->SetProperty("server path", "c:¥¥sa17¥¥bin32");
```

プロパティは、GetEvent メソッドによって返されるイベントのタイプを制御します。不要なイベントを無効にすることによって、パフォーマンスを向上できることがあります。イベントタイプは、対応するプロパティを 1 に設定すると有効になり、0 に設定すると無効になります。

次の表に、使用可能なプロパティ名と、各プロパティによって制御されるイベントタイプを示します。

プロパティ名	制御されるイベントタイプ	デフォルト値
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_INDEX	0
enable progress text	DBSC_EVENTTYPE_PROGRESS_TEXT	0
enable title	DBSC_EVENTTYPE_TITLE	0
enable sync start and done	DBSC_EVENTTYPE_SYNC_START DBSC_EVENTTYPE_SYNC_DONE	1

プロパティ名	制御されるイベントタイプ	デフォルト値
enable status	DBSC_EVENTTYPE_ML_CONNECT DBSC_EVENTTYPE_UPLOAD_COMMITTED DBSC_EVENTTYPE_DOWNLOAD_COMMITTED	1

## 関連情報

[StartServer\(unsigned, const char \\*, unsigned, DBSC\\_StartType \\*\) メソッド \[20 ページ\]](#)

[GetEvent\(DBSC\\_Event \\*\\*, unsigned\) メソッド \[13 ページ\]](#)

[GetProperty\(const char \\*, char \\*\) メソッド \[15 ページ\]](#)

[GetErrorInfo\(void\) メソッド \[13 ページ\]](#)

### 1.1.13 ShutdownServer(DBSC\_ShutdownType) メソッド

クライアントの接続先である dbmlsync サーバを停止します。

#### 構文

```
public virtual bool ShutdownServer (DBSC_ShutdownType how)
```

#### パラメータ

**how** サーバ停止の緊急度を示します。サポートされる値は DBSC\_ShutdownType 列挙にリストされます。

#### 戻り値

停止要求がサーバに正常に送信された場合は true、正常に送信されなかった場合は false を返します。false が返されたときは、GetErrorInfo メソッドを呼び出して、失敗に関する詳細な情報を取得できます。

#### 備考

Shutdown メソッドはすぐに戻りますが、サーバが実際に停止するまでに遅延が生じることがあります。

WaitForServerShutdown メソッドを使用すると、サーバが実際に停止するまで待つことができます。

#### i 注記

ShutdownServer を呼び出した後も、Disconnect メソッドを使用してください。

## 関連情報

[DBSC\\_ShutdownType 列挙 \[28 ページ\]](#)

[Disconnect\(void\) メソッド \[11 ページ\]](#)

[WaitForServerShutdown\(unsigned\) メソッド \[22 ページ\]](#)

[GetErrorInfo\(void\) メソッド \[13 ページ\]](#)

## 1.1.14 StartServer(unsigned, const char \*, unsigned, DBSC\_StartType \*) メソッド

指定したポートでまだ受信していない場合は、新しい dbmlsync サーバを起動します。

#### 構文

```
public virtual bool StartServer (
    unsigned port,
    const char * cmdline,
    unsigned timeout,
    DBSC_StartType * starttype
)
```

## パラメータ

**port** 既存の dbmlsync サーバがないかどうかをチェックする TCP ポート。サーバは、新しく起動される場合、このポートで受信するように設定されます。

**cmdline** dbmlsync サーバを起動するための有効なコマンドライン。コマンドラインには、次のオプションのみを含めることができます。これらのオプションは、dbmlsync ユーティリティに対して持つ意味と同じ意味を持ちます。-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。-c オプションは必ず指定します。

**timeout** dbmlsync サーバが起動された後、要求を受け入れる準備が完了するまでの最大待ち時間 (ミリ秒単位)。DBSC\_INFINITY を使用して応答を無期限に待機します。

**starttype** サーバが検出または起動されたかどうかを示すために設定される出力パラメータ。starttype がエントリ時に NULL 以外の値であり、かつ StartServer が true を返した場合、終了時に starttype が指している変数は DBSC\_StartType 列挙の値に設定されます。

## 戻り値

サーバがすでに実行されている場合、または正常に起動された場合は true、それ以外の場合は false を返します。false が返されたときは、GetErrorInfo メソッドを呼び出して、失敗に関する詳細な情報を取得できます。

## 備考

サーバがある場合、このメソッドは starttype パラメータを DBSC\_SS\_ALREADY\_RUNNING に設定し、その他の処理を行わずに戻ります。サーバが見つからない場合、cmdline 引数で指定されたオプションを使用して新しいサーバを起動し、そのサーバが要求を受け入れ始めるまで待つから、戻ります。

Windows Mobile デバイスでは通常、StartServer を正常に呼び出すには、先に *server path* プロパティを設定する必要があります。ただし、次の場合は *server path* プロパティを設定する必要はありません。

- アプリケーションが dbmlsync.exe と同じディレクトリにあります。
- dbmlsync.exe が Windows ディレクトリにあります。

## 関連情報

[GetErrorInfo\(void\) メソッド \[13 ページ\]](#)

## 1.1.15 Sync(const char \*, const char \*) メソッド

同期を実行するよう、dbmlsync サーバに要求します。

### 構文

```
public virtual DBSC_SyncHdl Sync (  
    const char * profile_name,  
    const char * extra_opts  
)
```

## パラメータ

**profile\_name** 同期のオプションを含む同期プロファイルの名前で、リモートデータベース内に定義されているものです。profile\_name が NULL の場合、プロファイルは使用されないため、extra\_opts パラメータに、同期に使用するすべてのオプションが指定されている必要があります。

**extra\_opts** 同期プロファイルのオプション文字列を定義するときと同じルールに従った形式の文字列。<オプション名>=<オプション値> 形式の要素のセミコロンで区切ったリストとして指定される文字列です。profile\_name が NULL

以外の場合、extra\_opts で指定したオプションは、profile\_name で指定した同期プロファイルにすでにあるオプションに追加されます。プロファイルに文字列のオプションがすでに存在する場合は、プロファイルにすでに格納済みの値が文字列の値に置き換わります。profile\_name が NULL の場合、extra\_opts に、同期に使用するすべてのオプションが指定されている必要があります。

## 戻り値

この同期要求をユニークに識別する DBSC\_SynchHdl 値を返します。この値は、クライアントがサーバから切断するまでの間でのみ有効です。エラーのために同期要求を作成できなかった場合は NULL\_SYNCHDL を返します。NULL\_SYNCHDL が返されたときは、GetErrorInfo メソッドを呼び出して、失敗に関する詳細な情報を取得できます。

## 備考

サーバに接続してから、このメソッドを呼び出してください。profile\_name と extra\_opts のうち少なくとも 1 つを NULL 以外の値にしてください。

戻り値は同期要求を識別し、要求のキャンセルや同期によって返されるイベントの処理に使用できます。

## 関連情報

[GetErrorInfo\(void\) メソッド \[13 ページ\]](#)

## 1.1.16 WaitForServerShutdown(unsigned) メソッド

サーバが停止したときかタイムアウトになったときのどちらか早い方で戻ります。

### 構文

```
public virtual bool WaitForServerShutdown (unsigned timeout)
```

## パラメータ

**timeout** サーバが停止するまでの最大待ち時間 (ミリ秒) を示します。DBSC\_INFINITY を使用して応答を無期限に待機します。

## 戻り値

サーバのシャットダウンによりメソッドが返された場合は true、それ以外の場合は false を返します。false が返されたときは、GetErrorInfo メソッドを呼び出して、失敗に関する詳細な情報を取得できます。

## 備考

WaitForServerShutdown を呼び出すには、先に ShutdownServer メソッドを呼び出す必要があります。

## 関連情報

[GetErrorInfo\(void\) メソッド \[13 ページ\]](#)

## 1.2 DBSC\_CancelRet 列挙

同期キャンセル試行の結果を示します。

### 構文

```
enum DBSC_CancelRet
```

## メンバー

メンバー名	説明	値
DBSC_CANCEL_OK_QUEUED	待機キュー内の同期をキャンセルしました。	1
DBSC_CANCEL_OK_ACTIVE	アクティブな同期をキャンセルしました。	2
DBSC_CANCEL_FAILED	同期をキャンセルできませんでした。	3

## 関連情報

[CancelSync\(DBSC\\_SyncHdl\) メソッド \(旧式\) \[7 ページ\]](#)

## 1.3 DBSC\_ErrorType 列挙

メソッド呼び出しが失敗した理由を示します。

### 構文

```
enum DBSC_ErrorType
```

### メンバー

メンバー名	説明	値
DBSC_ERR_OK	エラーは発生しませんでした。	1
DBSC_ERR_NOT_INITIALIZED	クラスが、Init メソッドを呼び出すことによって初期化されていません。	2
DBSC_ERR_ALREADY_INITIALIZED	すでに初期化されたクラスに対して、Init メソッドが呼び出されました。	3
DBSC_ERR_NOT_CONNECTED	dbmlsync サーバへの接続がありません。	4
DBSC_ERR_CANT_RESOLVE_HOST	ホスト情報を解決できません。	5
DBSC_ERR_CONNECT_FAILED	dbmlsync サーバへの接続が失敗しました。	6
DBSC_ERR_INITIALIZING_TCP_LAYER	TCP レイヤの初期化中にエラーが発生しました。	7
DBSC_ERR_ALREADY_CONNECTED	すでに接続が確立されているため、Connect メソッドが失敗しました。	8
DBSC_ERR_PROTOCOL_ERROR	これは内部エラーです。	9
DBSC_ERR_CONNECTION_REJECTED	dbmlsync サーバによって接続が拒否されました。  str1 は、サーバによって返された文字列を指します。この文字列には、接続試行が拒否された理由の詳細が含まれることがあります。	10
DBSC_ERR_TIMED_OUT	サーバからの応答を待っている間にタイムアウトになりました。	11
DBSC_ERR_STILL_CONNECTED	クラスがまだサーバに接続されているため、そのクラスに対して Fini を実行できませんでした。	12
DBSC_ERR_SYNC_NOT_CANCELED	サーバが同期要求をキャンセルできませんでした。同期がすでに進行中であることが原因と思われます。	14



メンバー名	説明	値
DBSC_ERR_INVALID_VALUE	SetProperty メソッドに無効なプロパティ値が渡されました。	15
DBSC_ERR_INVALID_PROP_NAME	指定したプロパティ名は無効です。	16
DBSC_ERR_VALUE_TOO_LONG	プロパティ値が長すぎます。プロパティは、DBCS_MAX_PROPERTY_LEN で指定されているバイト数よりも短くしてください。	17
DBSC_ERR_SERVER_SIDE_ERROR	sync のキャンセル中または追加中にサーバ側でエラーが発生しました。  str1 は、サーバによって返された文字列を指します。この文字列には、エラーの詳細が含まれることがあります。	18
DBSC_ERR_CREATE_PROCESS_FAILED	新しい dbmsync サーバを起動できません。	20
DBSC_ERR_READ_FAILED	dbmsync サーバからのデータの読み込み中に TCP エラーが発生しました。	21
DBSC_ERR_WRITE_FAILED	dbmsync サーバへのデータの送信中に TCP エラーが発生しました。	22
DBSC_ERR_NO_SERVER_RESPONSE	要求されたアクションを完了するために必要な応答をサーバから受信できませんでした。	23
DBSC_ERR_UID_OR_PWD_TOO_LONG	指定した UID または PWD が長すぎます。	24
DBSC_ERR_UID_OR_PWD_NOT_VALID	指定された UID または PWD が有効ではないか、接続するのに十分な権限が不足しています。	25
DBSC_ERR_INVALID_PARAMETER	関数に渡されたパラメータのいずれかが無効です。	26
DBSC_ERR_WAIT_FAILED	サーバが停止するのを待っている間にエラーが発生しました。	27
DBSC_ERR_SHUTDOWN_NOT_CALLED	ShutdownServer メソッドを呼び出す前に WaitForServerShutdown メソッドが呼び出されました。	28
DBSC_ERR_NO_SYNC_ACK	同期要求がサーバに送信されましたが、受信確認が受信されませんでした。サーバが要求を受信したかどうかを判別する方法はありません。  hdl1 は、送信された同期要求のハンドルです。サーバが要求を受信した場合は、このハンドルを使用すると、GetEvent を使用して取得された同期用イベントを識別できます。	29
DBSC_ERR_ACTIVE_SYNC_NOT_CANCELED	同期がアクティブであるため、サーバは同期をキャンセルできませんでした。	30

メンバー名	説明	値
DBSC_ERR_DEAD_SERVER	dbmlsync サーバで起動時にエラーが発生しました。  サーバを停止しています。dbmlsync の -o オプションを使用して、ファイルにエラーメッセージのログを記録してください。	31

## 1.4 DBSC\_EventType 列挙

同期によって生成されたイベントのタイプを示します。

### 構文

```
enum DBSC_EventType
```

### メンバー

メンバー名	説明	値
DBSC_EVENTTYPE_ERROR_MSG	同期によってエラーが生成されました。str1 はエラーのテキストを指します。	1
DBSC_EVENTTYPE_WARNING_MSG	同期によって警告が生成されました。str1 は警告のテキストを指します。	2
DBSC_EVENTTYPE_INFO_MSG	同期によって情報メッセージが生成されました。str1 はメッセージのテキストを指します。	3
DBSC_EVENTTYPE_PROGRESS_INDEX	進行状況バーを更新するための情報を提供します。val1 は、新しい進行状況値を格納します。  この値の範囲は 0 ~ 1000 で、0 は 0% 完了したことを示し、1000 は 100% 完了したことを示します。	4
DBSC_EVENTTYPE_PROGRESS_TEXT	進行状況バーに関連付けられているテキストが更新されました。新しい値は、str1 が指します。	6
DBSC_EVENTTYPE_TITLE	同期ウィンドウ/コントロールのタイトルが変更されました。新しいタイトルは、str1 が指します。	7
DBSC_EVENTTYPE_SYNC_START	同期が開始しました。このイベントに関連付けられている追加情報ははありません。	8

メンバー名	説明	値
DBSC_EVENTTYPE_SYNC_DONE	同期が完了しました。val1 は、同期からの終了コードを格納します。  0 という値は、成功を示します。0 以外の値は、同期が失敗したことを示します。	9
DBSC_EVENTTYPE_ML_CONNECT	Mobile Link サーバへの接続が確立されました。str1 は、使用されている通信プロトコルを示します。str2 は、使用されるネットワークプロトコルオプションを示します。	10
DBSC_EVENTTYPE_UPLOAD_COMMITTED	Mobile Link サーバは統合データベースへのアップロードを正常にコミットしたことを確認しました。	11
DBSC_EVENTTYPE_DOWNLOAD_COMMITTED	ダウンロードはリモートデータベースで正常にコミットされました。  val1 には、コミットされた挿入/更新操作の回数が含まれます。val2 には、コミットされた削除操作の回数が含まれます。	12
DBSC_EVENTTYPE_UPLOAD_START	リモートが、サーバへのアップロードの送信を開始しました。	13
DBSC_EVENTTYPE_UPLOAD_SENT	リモートが、サーバへのアップロードセグメントの送信を完了しました。  インクリメンタルアップロードとトランザクションアップロードの場合、アップロードセグメントが送信されるたびに別々のイベントが生成されます。val1 には、送信された挿入操作の数が含まれています。val2 には、送信された更新操作の数が含まれています。val3 には、送信された削除操作の数が含まれています。	14
DBSC_EVENTTYPE_DOWNLOAD_START	リモートが、サーバから受信したダウンロードの処理を開始しました。	15

## 関連情報

[DBSC\\_Event 構造体 \[31 ページ\]](#)

[GetEvent\(DBSC\\_Event \\*\\*, unsigned\) メソッド \[13 ページ\]](#)

## 1.5 DBSC\_GetEventRet 列挙

イベント取得の試行の結果を示します。

### 構文

```
enum DBSC_GetEventRet
```

### メンバー

メンバー名	説明	値
DBSC_GETEVENT_OK	イベントが正常に取得されたことを示します。	1
DBSC_GETEVENT_TIMED_OUT	返すことができるイベントがないまま、タイムアウトになったことを示します。	2
DBSC_GETEVENT_FAILED	エラーが発生したためにイベントが返されなかったことを示します。	3

### 関連情報

[GetEvent\(DBSC\\_Event \\*\\*, unsigned\) メソッド \[13 ページ\]](#)

## 1.6 DBSC\_ShutdownType 列挙

サーバを停止する緊急度を示します。

### 構文

```
enum DBSC_ShutdownType
```

## メンバー

メンバー名	説明	値
DBSC_SHUTDOWN_ON_EMPTY_QUEUE	サーバが未処理の同期要求を完了してから停止する必要があることを示します。 サーバは、停止要求を受信すると、それよりも後の同期要求を受け入れません。	1
DBSC_SHUTDOWN_CLEANLY	サーバができるだけ早く正常に停止する必要があることを示します。 未処理の同期要求は実行されません。実行中の同期は中断されることがあります。	2

## 関連情報

[ShutdownServer\(DBSC\\_ShutdownType\) メソッド \[19 ページ\]](#)

## 1.7 DBSC\_StartType 列挙

dbmlsync サーバを起動しようとしているときに実行されたアクションを示します。

### 構文

```
enum DBSC_StartType
```

## メンバー

メンバー名	説明	値
DBSC_SS_STARTED	新しい dbmlsync サーバが起動されたことを示します。	1
DBSC_SS_ALREADY_RUNNING	既存の dbmlsync サーバが見つかったため、新しいサーバが起動されなかったことを示します。	2

## 関連情報

[StartServer\(unsigned, const char \\*, unsigned, DBSC\\_StartType \\*\) メソッド \[20 ページ\]](#)

## 1.8 DBSC\_ErrorInfo 構造体

以前のメソッド呼び出しの失敗に関する情報が含まれています。

### 構文

```
typedef struct DBSC_ErrorInfo
```

### メンバー

DBSC\_ErrorInfo のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

#### 変数

変更子とタイプ	変数	説明
public DBSC_ErrorType	type	失敗の理由を示す値が含まれています。 サポートされる値は DBSC_ErrorType 列挙にリストされます。
public const char *	str1	失敗に関する詳細が含まれています。 この情報の意味は type 変数の値によって異なります。
public const char *	str2	失敗に関する詳細が含まれています。 この情報の意味は type 変数の値によって異なります。
public long int	val1	失敗に関する詳細が含まれています。 この情報の意味は type 変数の値によって異なります。
public long int	val2	失敗に関する詳細が含まれています。 この情報の意味は type 変数の値によって異なります。
public DBSC_SyncHdl	hdl1	失敗に関する詳細が含まれています。 この情報の意味は type 変数の値によって異なります。

## 備考

str1、str2、val1、val2、hdl1 には失敗に関する詳細が含まれ、その意味はエラータイプによって異なります。次のエラータイプはこの構造体のフィールドを使用して詳細を格納します。

- DBSC\_ERR\_CONNECTION\_REJECTED
- DBSC\_ERR\_SERVER\_SIDE\_ERROR
- DBSC\_ERR\_NO\_SYNC\_ACK

## 関連情報

[DBSC\\_ErrorType 列挙 \[24 ページ\]](#)

## 1.9 DBSC\_Event 構造体

同期によって生成されたイベントに関する情報が含まれています。

### 構文

```
typedef struct DBSC_Event
```

## メンバー

DBSC\_Event のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

### 変数

変数とタイプ	変数	説明
public DBSC_SyncHdl	hdl	イベントを生成した同期を示します。 この値は、Sync メソッドによって返される値と一致します。
public DBSC_EventType	type	レポートされるイベントのタイプを示します。
public const char *	str1	イベントに関する詳細が含まれています。 この情報の意味は type 変数の値によって異なります。

変更子とタイプ	変数	説明
public const char *	str2	イベントに関する詳細が含まれています。 この情報の意味は type 変数の値によって異なります。
public long int	val1	イベントに関する詳細が含まれています。 この情報の意味は type 変数の値によって異なります。
public long int	val2	イベントに関する詳細が含まれています。 この情報の意味は type 変数の値によって異なります。
public long int	val3	イベントに関する詳細が含まれています。 この情報の意味は type 変数の値によって異なります。
public void *	data	イベントに関する詳細が含まれています。 この情報の意味は type 変数の値によって異なります。

## 関連情報

[DBSC\\_EventType 列挙 \[26 ページ\]](#)



---

## 2 このマニュアルの印刷、再生、および再配布

次の条件に従うかぎり、このマニュアルの全部または一部を使用、印刷、再生、配布することができます。

1. ここに示したものとそれ以外のすべての著作権と商標の表示をすべてのコピーに含めること。
2. マニュアルに変更を加えないこと。
3. SAP 以外の人間がマニュアルの著者または情報源であるかのように示す一切の行為をしないこと。

ここに記載された情報は事前の通知なしに変更されることがあります。

# 重要免責事項および法的情報

## コードサンプル

この文書に含まれるソフトウェアコード及び / 又はコードライン / 文字列 (「コード」) はすべてサンプルとしてのみ提供されるものであり、本稼働システム環境で使用することが目的ではありません。「コード」は、特定のコードの構文及び表現規則を分かりやすく説明及び視覚化することのみを目的としています。SAP は、この文書に記載される「コード」の正確性及び完全性の保証を行いません。更に、SAP は、「コード」の使用により発生したエラー又は損害が SAP の故意又は重大な過失が原因で発生させたものでない限り、そのエラー又は損害に対して一切責任を負いません。

## アクセシビリティ

この SAP 文書に含まれる情報は、公開日現在のアクセシビリティ基準に関する SAP の最新の見解を表明するものであり、ソフトウェア製品のアクセシビリティ機能の確実な提供方法に関する拘束力のあるガイドラインとして意図されるものではありません。SAP は、この文書に関する一切の責任を明確に放棄するものです。ただし、この免責事項は、SAP の意図的な違法行為または重大な過失による場合は、適用されません。さらに、この文書により SAP の直接的または間接的な契約上の義務が発生することは一切ありません。

## ジェンダーニュートラルな表現

SAP 文書では、可能な限りジェンダーニュートラルな表現を使用しています。文脈により、文書の読者は「あなた」と直接的な呼ばれ方をされたり、ジェンダーニュートラルな名詞 (例:「販売員」又は「勤務日数」) で表現されます。ただし、男女両方を指すとき、三人称単数形の使用が避けられない又はジェンダーニュートラルな名詞が存在しない場合、SAP はその名詞又は代名詞の男性形を使用する権利を有します。これは、文書を分かりやすくするためです。

## インターネットハイパーリンク

SAP 文書にはインターネットへのハイパーリンクが含まれる場合があります。これらのハイパーリンクは、関連情報を見いだすヒントを提供することが目的です。SAP は、この関連情報の可用性や正確性又はこの情報が特定の目的に役立つことの保証は行いません。SAP は、関連情報の使用により発生した損害が、SAP の重大な過失又は意図的な違法行為が原因で発生したものでない限り、その損害に対して一切責任を負いません。すべてのリンクは、透明性を目的に分類されています (<http://help.sap.com/disclaimer> を参照)。



[go.sap.com/registration/  
contact.html](http://go.sap.com/registration/contact.html)

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

本書のいかなる部分も、SAP SE 又は SAP の関連会社の明示的な許可なくして、いかなる形式でも、いかなる目的にも複製又は伝送することはできません。本書に記載された情報は、予告なしに変更されることがあります。SAP SE 及びその頒布業者によって販売される一部のソフトウェア製品には、他のソフトウェアベンダーの専有ソフトウェアコンポーネントが含まれています。製品仕様は、国ごとに変わる場合があります。

これらの文書は、いかなる種類の表明又は保証もなしで、情報提供のみを目的として、SAP SE 又はその関連会社によって提供され、SAP 又はその関連会社は、これら文書に関する誤記脱落等の過失に対する責任を負うものではありません。SAP 又はその関連会社の製品及びサービスに対する唯一の保証は、当該製品及びサービスに伴う明示的な保証がある場合に、これに規定されたものに限られます。本書のいかなる記述も、追加の保証となるものではありません。

本書に記載される SAP 及びその他の SAP の製品やサービス、並びにそれらの個々のロゴは、ドイツ及びその他の国における SAP SE (又は SAP の関連会社) の商標若しくは登録商標です。本書に記載されたその他のすべての製品およびサービス名は、それぞれの企業の商標です。

商標に関する詳細の情報や通知については、<http://www.sap.com/corporate-en/legal/copyright/index.epx> をご覧ください。