New Features in PowerBuilder 12.0

Click on a topic for more information, then use the Browse buttons on the Help window to navigate:

- PowerBuilder Classic and PowerBuilder .NET
- Enhancements for ADO.NET 1.1 Providers
- Enhancements for ADO.NET 2.0 Providers

PowerBuilder Classic and PowerBuilder .NET

Beginning with the 12.0 release, PowerBuilder installs with two separate IDEs. The familiar PowerBuilder IDE is rebranded as PowerBuilder Classic. The new IDE is named PowerBuilder .NET.

The PowerBuilder Classic IDE retains the same basic functionality as in earlier PowerBuilder releases. The PowerBuilder .NET IDE hosts the Visual Studio isolated shell and is designed for full compliance with the common language specifications for .NET.

PowerBuilder .NET includes one new target (WPF Window Application) and two new projects (WPF Window Application and WCF Client Proxy). The WCF Client Proxy is not enabled in the CTP release. Also after the CTP release, the .NET Web Services and .NET Assembly targets will be ported to PowerBuilder .NET, where you can take advantage of the language enhancements for fuller .NET compliance. Although the wizards for these targets are currently visible in the PowerBuilder .NET IDE, they are not yet functional.

For information about PowerBuilder .NET targets and projects, see the *PowerBuilder .NET Features Guide*.

Enhancements for ADO.NET 1.1 Providers

The PowerBuilder ADO interface provides design-time and run-time support for connecting to specific ADO.NET 1.1 providers. The following drivers have been updated to support new database features of their providers.

- Oracle Data Provider
- ASE ADO.NET
- Microsoft SQL Server Features

Oracle Data Provider

This section describes PowerBuilder changes supporting Oracle Data Provider for .NET (ODP.NET) connections.

ODP.NET Driver Updates

Drivers for these ODP.NET versions are updated:

- For Oracle 10g, the *Oracle.DataAccess.Client.dll* driver was upgraded from Version 10.1.0.301 to 2.102.2.20.
- For Oracle 11, the *Oracle.DataAccess.Client.dll* Version 2.111.6.20 driver was added.

Both drivers are ADO.NET compatible.

Note The PowerBuilder ADO.NET interface no longer includes a driver for Oracle ODP.NET 9i. Users of that provider should migrate to Oracle ODP.NET 10g.

New Features for ADO.NET 10.2 and Earlier

The following new features are supported:

- **Client Identifier:** The client identifier is a predefined attribute for the Oracle application context namespace, USERENV. Like proxy authentication, the client identifier enables tracking user identities. However, unlike proxy authentication, the client identifier does not require separate sessions for the proxy user and end user. Also, the client identifier does not need to be a database user, and can be set to any string. Most important, the client identifier enables ODP.NET developers to use application context and Oracle Label Security, and to configure an Oracle Virtual Private Database (VPD) more easily.

  Configure the client identifier for Oracle ADO.NET data providers in the Driver Specific tab of the Database Profile Setup dialog.

- **Connection Pool Optimizations for RAC Databases:** An Oracle Data Provider for ADO.NET optimizes connection pooling for Real Application Cluster (RAC) databases by balancing work requests across Oracle RAC instances, based on load balancing advisory and service requirements. In addition, the ODP.NET connection pool can be enabled to proactively free resources associated with connections that have been severed when an Oracle RAC service, instance, or node goes down.

  Specify ODP.NET connection pool optimizations as arguments to the ProviderString DBParm parameter. You can enter driver-specific parameters at the bottom of the Connection tab of the Database Profile Setup dialog.

- **Large Object Retrieval:** You can retrieve entire columns of large object (LOB) data even if the select list does not contain a primary key, row id, or unique key. To use this enhancement, set the InitialLOBFetchSize property value to `-1` for CLOB and BLOB objects.

- **LONG Retrieval:** You can retrieve entire columns of LONG and LONGRAW data even if the select list does not contain a primary key, row id, or unique key. To use this enhancement, set the InitialLONGFetchSize property value to `-1`.

- **XMLType:** The Oracle XMLType datatype is mapped to the PowerBuilder string type, with these limitations:

  - XMLType cannot be used in Where clauses within PowerBuilder Embedded SQL statements or in DataWindow.

  - XMLType columns cannot be selected directly by an Oracle cursor.

    For example:

    ```
    CREATE OR REPLACE Function p_Ora_sp_char_11
    return types.cursortype
     AS
     l_cursor    types.cursorType;
     begin
     open l_cursor for select col1 from
    t_Ora_sp_char_11;
     return l_cursor;
     end;
    ```

    To use the preceding statement in PowerBuilder, modify it to:

    ```
    CREATE OR REPLACE Function p_Ora_sp_char_11
    return types.cursortype
     AS
     l_cursor    types.cursorType;
    ```

```
 begin
 open l_cursor for select x.col1.getstringval()
from t_Ora_sp_char_11 x;
 return l_cursor;
 end;
```

- XMLType cannot be a parameter of a procedure or function, because PowerBuilder binds XMLType as a string type, but Oracle does not support that usage. The following SQLPLUS statement illustrates:

```
declare
ss varchar2(1000);
BEGIN
ss := '<Date>1994-08-07</Date>';
mysp2(ss);
END;
```

- **Client Access Through a Proxy:** With proxy authentication, the end user typically authenticates to a middle tier (such as a firewall), that in turn logs into the database on the user's behalf, as a proxy user. After logging into the database, the proxy user can switch to the end user's identity and perform operations using the authorization accorded to that user.

  The Connection tab of the Database Profile Setup dialog provides a Connect As dropdown control. To create a proxy connection, enter a different value that is not one of the predefined control items (Default, SYSOPER, and SYSDBA).

- **Transparent Application Failover Notification:** Transparent Application Failover (TAF) notification enables an application connection to automatically reconnect to another database instance if the connection is severed. When a failover occurs, applications may wish to be notified.

  A new DBParm, SvrFailover, supports TAF notification. By default, SvrFailover is set to 0. If SvrFailover is set to yes, true, or 1 dbnotification event in transaction object will be invoked when TAF occurs.

- **Microsoft ADO.NET 2.0:** The driver for ODP.NET 10 is ADO.NET 2.0-compliant.

- **Database Change Notification Support:** ODP.NET provides a notification framework that supports Continuous Query Notification. It enables applications to receive notifications when there is a change in a query result set, schema objects, or the state of the database. By using Continuous Query Notification, an application can easily maintain the validity of the client-side cache (for example, the ADO.NET DataSet).

The following new feature is *not* supported:

Connection Pooling Management: Oracle Data Provider for .NET connection pool management provides explicit connection pool control to ODP.NET applications. Applications can explicitly clear connections in a connection pool or all the connection pools.

PowerBuilder does not support explicitly clearing connections in a connection pool.

New Features for ADO.NET 1.1

The following .NET 1.1 features are supported:

- **ODP.NET Configuration:** Developers can now configure ODP.NET using configuration files, including the .NET application configuration file, *web.config*, and *machine.config*. Settings in the machine.config file override the registry settings. The settings in the application configuration fileor the *web.config file* overrides the values in the *machine.config* file.
- **Additional Connection Pool Optimizations for RAC and Data Guard:** ODP.NET now cleans up the connection pool when the database down event is received from Real Application Clusters (RAC) or Oracle Data Guard. This is in addition to the events for which ODP.NET previously cleaned up the connection pool: node down, service member down, and service down.
- **Windows-Authenticated User Connection Pooling:** You can now manage operating system-authenticated connections as part of ODP.NET connection pools, through Windows account management.
- **Connection Pool Performance Counters:** ODP.NET publishes performance counters for connection pooling, which can be viewed using the Windows Performance Monitor.

  For PowerBuilder, the counters can be set in the Windows registry or in the application configuration file.

The following ADO.NET 1.1 features are *not* supported:

- **Oracle User-Defined Types:** PowerBuilder does not support UDT types.
- **Bulk Copy Operations:** ADO.NET 1.1 enables applications to efficiently load large amounts of data from a table in one database to another table in the same or a different database.

  PowerBuilder does not support bulk copies; instead it uses pipelines for table copy operations.

ASE ADO.NET

This section describes changes for connections to Adaptive Server Enterprise ADO.NET database providers.

ADO.NET Driver update

Drivers for these ADO.NET versions are updated:

- The ASE 12.5*x* ADO.NET driver, *Sybase.Data.AseClient.dll*, is updated from Version 1.1.411.0 to 1.1.670.0.

  The ASE 12.5*x* ADO.NET driver is ADO.NET 1.1 compatible, and does not support ADO.NET 2.0.

- The ASE 15 ADO.NET driver is updated from *Sybase.Data.AseClient.dll* Version 1.15.50.0 to *Sybase.AdoNet2.AseClient.dll* 1.15.325.0.

  The ASE 15 ADO.NET driver is ADO.NET 2.0 compatible.

New Features for ASE 15

The ASE 15 ADO.NET driver supports these new ASE identity types:

bigint
int
smallint
tinyint
unsigned bigint
unsigned int
unsigned smallint

Microsoft SQL Server Features

This section describes changes to the PowerBuilder ADO interface for Microsoft SQL Server:

New Features for SQL Server 2005 and Earlier

The following new features are supported:

- Large value types:

  varchar(max)
  nvarchar(max)
  varbinary(max)

  xml, varchar(max) and nvarchar(max) are mapped to the PowerBuilder string type; varbinary(max) is mapped to the PowerBuilder blob type.

- PowerBuilder supports SQL Server database mirroring, and a DBNotification event is fired when failover occurs.

A new DBParm parameter, FailoverPartner, enables you to set the SQL Server failover partner server, as in the SQL Native Client (SNC) interface.

Query notifications are *not* supported by the PowerBuilder ADO interface for SQL Server.

New Features for SQL Server 2008

The following SQL Server 2008 features are supported:

- New datatypes:

  date
  time
  datetime2
  varbinary(max) (filestream)

  The SQL Server date, time and datetime2 datatypes are mapped to PowerBuilder date, time and datetime types. varbinary(max) (filestream) is mapped to the PowerBuilder blob type. The maximum scale of time or datetime2 is 6.

- MERGE statement
- Grouping sets
- Row constructors
- Table hints
- The new T-SQL command works in the PowerBuilder ADO interface for SQL Server, as in the SNC interface.

These SQL Server 2008 features are *not* supported:

- datetimeoffset datatype
- Table-valued parameters

Enhancements for ADO.NET 2.0 Providers

In addition to provider-specific enhancements for ADO.NET 1.1 connections, the PowerBuilder 12 ADO interface has been extended to support NET 2.0 providers. Now, you can connect at run time to any data source that adheres to the .NET 2.0 Common Provider model.

The Database Profile Setup dialog automatically detects and lists all ADO.NET providers on the system. Otherwise, this dialog is unchanged from the previous version, enabling you to define profiles for each provider in the same way.

At run time, the user specifies the provider name defined in the namespace DBParm. If PowerBuilder fails to load the provider it returns an error.

Provider-Specific Design-Time Support

While enabling connections to any ADO.NET 2.0 compliant provider at run time, the PowerBuilder IDE also includes design-time support for these specific ADO.NET 2.0 providers:

- ADO.NET for Adaptive Server Anywhere

  Connect to an ASA database using an iAnywhere.Data.SQLAnywhere provider. PowerBuilder applications can perform all database related operations, such as exploring ASA database objects like tables and procedures, and retrieving and updating data in the Database Painter.

- IBM.Data.Informix

  Access an Informix database using the IBM.Data.Informix provider (Informix client SDK version 3.5 or above). You can explore an Informix Database the Database Painter and perform all database related operations.

  Note The Informix DATETIME HOUR TO SECOND type is treated as type TIME in PowerBuilder. Also, the TIME type column is displayed in the Database Painter as DATETIME, because the two variants of Informix DATETIME type are indistinguishable in the resultset schema.

- ADO.NET for DB2

  PowerBuilder supports DB2 using the System.Data.Odbc provider for both runtime and design time operations.

Limitations

Except for providers that PowerBuilder specifically supports, database features for .NET 2.0 providers are limited to those supported by the ADO.NET 2.0 common library.

If a provider does not adhere to all the ADO.NET 2.0 specification and if it does not supply required data such as schema information, some design-time features in the Database Painter might not work properly.

Common Provider classes defined in ADO.NET 2.0 provide access only to basic functionality. Schema information returned by database providers is provider-specific, and the schema format is not defined in the ADO.NET common provider APIs.

PowerScript Syntax

To connect to a database, specify the profile namespace in the DBParm parameter. As with other database drivers, you can simply copy the SQLCA definition from the Database Profile Setup dialog and paste it in your PowerScript object.

From the SQLCA parameters and DBParms, PowerBuilder generates a connection string in the appropriate format to connect to a given provider's database. PB can create connection string for the existing providers Oracle, ASE, SQLServer, and the newly supported ASA, Informix and DB2 providers.

For example, the DataSource DBParm is translated to DSN=value for ODBC, and Server=value for Informix and DB2 providers. To use a different provider or connection string parameter, provide a partial or complete connection string in the Driver-Specific Parameters field (the PROVIDERSTRING DBParm).

Here are some examples of SQLCA syntax for connections to ASA, Informix, and DB2 providers:

- ASA

```
SQLCA.DBMS = "ADO.Net"

SQLCA.LogPass = <***>

SQLCA.LogId = "dba"

SQLCA.AutoCommit = False

SQLCA.DBParm =
"Namespace='iAnywhere.Data.SQLAnywhere',DataSource=
'SQL Anywhere 11 Demo'"
```

- Informix

```
SQLCA.DBMS = "ADO.Net"

SQLCA.LogPass = <***********>

SQLCA.LogId = "myId"

SQLCA.AutoCommit = False

SQLCA.DBParm =
"Namespace='IBM.Data.Informix',Database='my_db',Dat
```

```
aSource='informix11'"
```

Note The Informix server name is configured in the Informix client. To access Unicode data, set both db_locale (in the driver-specific PROVIDERSTRING DBParm) and the client_locale to en_us.utf8.

Example:

```
SQLCA.DBParm =
"Namespace='IBM.Data.Informix',Database='my_db',Dat
aSource='informix11',
PROVIDERSTRING='db_locale=en_us.utf8;
client_locale=en_us.utf8;'"
```

- DB2 (using odbc provider)

```
SQLCA.DBMS = "ADO.Net"

SQLCA.LogPass = <********>

SQLCA.LogId = "sa"

SQLCA.AutoCommit = False

SQLCA.DBParm =
"Namespace='System.Data.Odbc',DataSource='DB2_DSN'"
```