Integrating DataDirect Cloud and Progress Easyl into OpenEdge using the ODBC Bridge Sample Applications

Document version 2.0

Peter Judge & David
Moloney
Progress Software Corp.
February 2014

Contents

Introduction

OpenEdge applications are often deployed alongside other business systems that are responsible for key pieces of the overall business process. In these scenarios it is imperative that OE also connect (or integrate) with these other applications in order to execute the business process. DataDirect Cloud is a tool that facilitates this integration with SaaS applications like Salesforce, Microsoft Dynamics CRM, Oracle RightNow, and a number of other sources in the development pipeline. More information about DataDirect Cloud is available at http://www.datadirectcloud.com/why-datadirect-cloud.html . Easyl is a tool that also aggregates data from many data sources but also allows mapping and reporting options to enable data collaboration in the cloud. More information about Easyl is available at http://www.progress.com/products/easyl.

This document represents the second in a series of sample publications expected using the “ODBC Bridge”. The ODBC Bridge is simply the ODBC API linked into ABL applications for purposes of gaining connectivity to ODBC data sources (DSN’s). The first sample published was built to certify against the DataDirect Cloud ODBC drivers and can be found here: https://community.progress.com/technicalusers/w/openedgecloudarcade/2089.integrating-a-datadirect-cloud-server-into-openedge-applications.aspx.

This document represents the second publication and incorporates samples and certification for both ODBC drivers from DataDirect Cloud and the Easyl Client Application. The samples themselves are more or less the same as before with some cleanup on error handling. The new project also sets up samples for executing both DataDirect Cloud and Easyl and includes added tips for getting started with Easyl.

These two sample ABL applications use the ODBC Bridge to show how you can interact with the DataDirect Cloud (D2C) server and Easyl services, respectively, using a set of OOABL classes. These classes provide a simplified wrapper around a series of DLL calls to the D2C Driver via the ODBC Driver Manager. This simplification is primarily around removing the need to work directly with the ODBC API via memptrs and the like. The classes simplify the calls a consumer of the API must make in order to execute a SQL statement, gather corresponding metaschema and data results. By encapsulating the process, the ABL programmer needs only to formulate a SQL query and designate a target for its results while the classes process all the handle allocations, state transitions and cleanup.

The API returns the data from the query executed (the result set) as either a JSON Object, or a dynamically-created temp-table. This temp-table is created from the column information of the query.

The sample code allows you to connect to any D2C data source or Easyl data set and to query D2C data sources or Easyl data set objects in any way you see fit.

Setup & environment

Prerequisites

- The API assumes that you are running a version of OE that has the same bitness as the underlying OS (ie 32-bit OE on 32-bit Windows). The default OpenEdge Windows install is a 32-bit install, although as of 11.3.0 there will be a 64-bit version. Trying to run 32-bit OpenEdge with the 64-bit drivers results in an error. There are some tweaks needed to install the 32-bit DLLs on a 64-bit OS, like Windows 7; these tweaks involve the registry settings and won't be described here.

- The fact that the datasources ultimately reside on the public internet somewhere means that you will need to be able to access public sites such as DataDirectCloud.comand, easyl.progress.com and data source sites such as salesforce.com.

Setting up (cloud) data sources

There are a few setup steps required to be able to access cloud data sources. The DataDirectCloud setup walks through setting up a salesforce.com data source while the Easyl platform setup walks through setting up an on-premise OpenEdge database in the cloud. Neither data source configuration is particularly compelling in itself and is provided only as a means to achieve preliminary success with the OpenEdge sample applications. For instance, once you set up an on-premise OE database as an Easyl data source, consider configuring other data sources, like salesforce.com (described within), so that you can combine other information into your Easyl data set. Combining data from disparate data sources to produce a meaningful conjunction of cloud information in a common data set is some of the rich power of Easyl.

Set up an account on salesforce.com or another supported cloud provider in DataDirectCloud or the Easyl Platform

This application utilizes data source name (DSN) configurations from the ODBC Administrator that redirect database request to your real data sources, i.e., locations where the data actually lives, using indirection from the cloud. NOTE: Some providers have time-limited accounts for trial purposes (eg Salesforce.com has a 30-day limited account).

⚠ Most of the DataDirect cloud examples in this document are based on a Salesforce.com datasource, since it was the first data source provided by DataDirect Cloud. The examples are also Windows-based (pertinent for setup more than runtime).

⚠ The Easyl example described in this document is based on an initial implementation of Easyl using a simple, on-premise, OpenEdge data source. Further instruction on setting up this Easyl data set can be found in the /doc directory of the download entitled, “Including your on-premise OpenEdge database into an Easyl Data Set”.

Create an account on the DataDirect Cloud site

Register for an account on http://www.DataDirectCloud.com.
NOTE: Implicit in your DataDirect Cloud setup is the fact that you will need to have obtained a Progress ID. The need for a ProgressID is also a requirement for logging into Easyl at https://login.easyl.progress.com and for downloading the Easyl Client Application.

Set up a cloud data source on DataDirect Cloud

Easyl data source configurations are utilized with templates from Progress Pacific Easyl platform. For documentation on how to setup data sources, templates, data sets and reports in Easyl, go to http://documentation.progress.com/output/DataDirect/Easyl/ . Easyl cloud data source configuration is essentially the same as for the DataDirect Cloud.

For setting up a Salesforce data source for DataDirect cloud connectivity, create a new data source in your DataDirectCloud account by selecting the Salesforce.com data source.

Setting	**Value**
Username	This will be your Salesforce.com login
Password	Your super-secret password.
Salesforce Login URL	`http://login.salesforce.com`

Setting	Value
Security token	The Salesforce.com requires a token for remote access. This value **must** be specified in the Advanced tab.
Data Source Name	<any value>. This value will be used for the ODBC database name. For example, **`sfDSN`**

Download and install the D2C drivers

In DataDirectCloud, select the correct ODBC drivers for your system from the selection provided on the Downloads tab.

For Easyl, the D2C drivers are bundled with the Easyl Client Application and can be obtained by downloading this software to the machine from which you will run the sample application.

> ℹ The **ODBC Drivers Readme** talks about needing to install the Microsoft MDAC software. This is not necessary on Windows 7.

Set up a local data source for DataDirect Cloud configuration

In the Windows **ODBC Administrator** tool, add a System DSN (or a User DSN, the setup is the same).

Setting	Notes
Data Source Name	<any value>. You can use something like **OED2C**
Database Name	The value specified above for the salesforce data source's name (see above). Eg, **`sfDSN`**
Authentication: User Name	Your user name (Progress ID) for DataDirectCloud
Authentication: Login Domain	<blank>
Data Source Authentication: Data Source User	<blank>

Setting	Notes
Data Source Authentication: Data Source Password	<blank>

Test the connection. You will be prompted for the password of your Progress ID for your DataDirectCloud account.

Set up a local data source for Easyl configuration

In the Windows **ODBC Administrator** tool, add a System DSN (or a User DSN, the setup is the same).

Setting	Notes
Data Source Name	<any value>. You can use something like **OEEasyl**
Database Name	This needs to be equal to the name of the data set you configured on the Easyl Cloud Platform.
Authentication: User Name	Your user name (Progress ID) for Easyl
Service	“Easyl” This is the name of the service you will use to interact with Easyl Platform services. The Service defaults to “Easyl” when you download the Easyl Client Application.

Test the connection. You will be prompted for the password of your Progress ID for your Easyl account.

Add test data

For DataDirect Cloud, you will need to add some test data to your data sources in order to be able to query them. Some data sources may provide this test data for you.

For Easyl, you can go to the the document entitled “Including your on-premise OpenEdge database into an Easyl Data Set” to set up a single on-premise data source for Easyl and just use data from your on-premise OpenEdge database for

Easyl client access. This does not, in itself, demonstrate the power of Easyl but can provide initial configuration success so that Easyl can be further explored. You are encouraged to expand your data source access for Easyl in the same way you would add data sources for the DataDirect cloud. However, Easyl requires the additional step of adding these data sources to templates and data sets so they can be mapped to Easyl business objects and accessed through the Easyl ODBC driver from Easyl clients, like OpenEdge.

Setting up the OpenEdge environment

Download the ABL API

The DataDirect Cloud sample application ("run_D2C_sample") was developed against OpenEdge release 11.3, and depends on a procedure library (PL) file shipped with that release. The Easyl platform was integrated into the Easyl sample application ("run_EASYL_sample") starting in OpenEdge release 11.3.3. You can begin to use the DataDirect Cloud and Easyl sample applications with the procedure library (PL) that ships in OpenEdge release 11.3. The D2C sample is forward-compatible from the 11.3.2 release. The Easyl sample is forward-compatible from the 11.3.3 release.

The ABL API samples for DataDirect Cloud are available via the Samples page in Progress Developer Studio for OpenEdge (PDSOE). It can be accessed via the **Help > Samples** menu. The Easyl sample is not yet available from the **Help > Samples** menu but both samples are also available via Progress Communities at https://community.progress.com. This document you are reading appears in the **doc** folder of the DataDirectCloud_EASYL_ABL API sample application project. NOTE: The predecessor project that only included a sample for DataDirectCloud was named DataDirectCloud_ABL API. Since the new project is inclusive of the old project, you may want to replace the old with the new (assuming you do not have edits to the old project you wish to keep).

To manually create the project, use PDSOE's **File > Import** functionality and select the **`abl_d2c_easyl.zip`** archive. A project called `DataDirectCloud_EASYL_ABL_API` will be created for you.

Note that the DataDirectCloud sample in the project requires no local database, since all data will be retrieved from the Cloud data source.

For the Easyl application sample in the project, you can use Cloud data sources you configure on the Easyl platform or you can start by just configuring a local on-premise OpenEdge database to be the data source target of your Easyl data Set. For more information on performing your initial Easyl setup with an on-premise OpenEdge database, locate the document in the /doc folder entitled, “Including your on-premise OpenEdge database into an Easyl Data Set”.

The project contains 2 Launch Configurations called `Run D2C Sample.launch` for the DataDirectCloud sample application which runs the sample procedure, `test/run_D2C_sample.p` and the `Run EASYL Sample.launch` for the Easyl sample application which runs the sample procedure, `test/run_EASYL_sample.p`. The code snippets in this document are taken from these procedures.

Dependencies

If you choose not to use PDSOE, you will need to manually change your propath in order to run the sample. On Windows, add `%DLC%/gui/rules/OpenEdge.BusinessRules.pl` to the propath. `%DLC%` represents the location of your OpenEdge 11.3 installation. If you would like to use this code in an AppServer environment, change the `'gui'` to `'tty'` in the propath entry.

Running the sample

The samples provided connect to a D2C data source or an Easyl data set and execute a couple of queries against them.

Connection

The first thing that the ABL needs to do is to connect to the D2C server or the Easyl dataset. The connection is done using the `OpenEdge.Core.ServerConnection.IServerConnection` infrastructure created for connecting to a Business Rules server that was added to OE 11.3 for D2C and OE 11.3.2 for Easyl.

Configure the ABL with your username/etc

The connection information is contained in a file in JSON format, in the `cfg` folder in **`d2c.json`** for DataDirectCloud access and **`easyl.json`** for Easyl access. Replace the 'null' values below with the DataSourceName, user name and password you used to set up the System DSN above. NOTE: In the Easyl data source name

(DSN) configuration, ensure that the database name matches the name of the data set you configured on the Easyl platform.

```
{ "DataSourceName"    : null,
  "UserName"          : null,
  "Password"          : null }
```

Connection code

The snippet below highlights the connection to the D2C server, via the `OpenEdge.Core.ServerConnection.ODBCConnection` object. This object is the interface through which we interact with the D2C server or Easyl services.

```
oConfig = cast(
    new ObjectModelParser():ParseFile('cfg/d2c.json'),
           JsonObject).
oD2CServer = new ODBCConnection(oConfig).
oD2CServer:Initialize().
```

Once we've connected to the server, and initialised it, we can query the data within.

Direct query execution

The D2C connection now allows us to perform SQL queries against the data source defined in D2C. The queries shown in the sample application are against Salesforce.com; obviously the tables and fields queried will be specific to each data source.

The Easyl connection allows us to perform SQL queries against the business objects defined in the Easyl data set. The queries show in the sample application are against no particular business object. The format of the select looks as follows:

```
cStmt = ’select “attribute1”, “attribute2”, attribute3”
from “object1”’.
```

It is important to double quote the attribute and object names to ensure they are translated properly. The “attributes” are just data columns mapped into a particular business object and the “object” is the collection of attributes stored in one named business object. Easyl can have one to many business objects defined in an Easyl data set in which a “data set” can be considered as the equivalent of a database. Obviously the analogy breaks down when one considers that each object and each data set can be mapped to many Easyl data sources.

If you use the document entitled “Integration your on-premise OpenEdge database into an Easyl Data Set” to set up a single on-premise data source for Easyl, you could consider configuring your Easyl Data Set based on an OpenEdge Sports database for your sample application. In that case, you could have chosen familiar attributes such as “Cust-Num” and “Name” from the “Customer” table object and mapped them to a business object named “cust-obj”. Now, when you create a select to run against your Easyl data set for on-premise OpenEdge sports database, you might formulate a statement such as the following:

```
cStmt = ’select “Cust-num”, “Name” from “cust-obj”’.
```

Retrieving schema information

There are 2 methods on the `OpenEdge.Core.ServerConnection.ODBCConnection` type for getting table and related schema information from the data source. These are **`GetTableSchema`** and **`GetTables`**. The former is for a single table, the latter for all tables. More detail appears later in this document.

It is not necessary to retrieve the schema in order to execute a query (ie if you already know the table/field names).

Returning a temp-table

The API will execute the passed-in SQL statement, using the query’s result set to create and populate an ABL temp-table named `resultset`. The temp-table will contain a field for each column in the query, and a record for each row in the SQL result set that is returned.

Currently the API is limited to returning a dynamically-created temp-table (as opposed to populating a pre-existing temp-table). This means you need to use dynamic queries etc to work with the data.

‘

```
/* Execute a SQL SELECT statement and get the result set as an ABL
temp-table */
cStmt = " select USERNAME, LASTNAME, EMAIL from USER ".

oD2CServer:ExecuteStatement(
             input cStmt,
             output table-handle hResultSet).

/* just dump the output to disk. Obviously you would do more with
this data in the real world */
hResultSet:write-json(
           'file', 'temp/resultset-d2c-table.json', true).
```

Note: Using the run_EASYL_sample, the “write-json” expression writes to a file named, “temp/resultset-easyl-table.json” instead.

Returning JSON-formatted data

The API will execute the passed-in SQL statement, and return an ABL **`JsonObject`**

```
/* Execute a SQL SELECT statement and get the result set in JSON
form */
cStmt = "select ACCOUNTNUMBER, SYS_NAME, ANNUALREVENUE,
NUMBEROFEMPLOYEES, DESCRIPTION, SLAEXPIRATIONDATE from ACCOUNT ".

oD2CServer:ExecuteStatement(input cStmt, output oResultSet).

/* just dump the output to disk. Obviously you would do more with
this data in the real world */
if valid-object(oResultSet) then
    oResultSet:WriteFile('temp/resultset-d2c-json.json', true).
```

based on the results of that query. This JSON data will contain both the schema information about the fields in the query, and also the data returned in the SQL result set.

The JSON object returned has 3 properties: `query` (the query that was executed), `columns` (schema information about the columns used by the query) and `resultset` (the data returned). The (truncated) output from the query about is shown below

Note: Using the run_EASYL_sample, the “write-json” expression writes to a file named, “temp/resultset-easyl-json.json” instead.

JSON result set

```
{"query"        : "select ACCOUNTNUMBER, SYS_NAME, ANNUALREVENUE,
          NUMBEROFEMPLOYEES, DESCRIPTION, SLAEXPIRATIONDATE from
          ACCOUNT ",
  "columns"     : [
     {"ColumnNum"     : 1,
      "ColumnName"    : "ACCOUNTNUMBER",
      "CType"         : 1,
      "CTypeSize"     : 1025,
      "AblType"       : "CHARACTER",
      "NumDecimals"   : 0,
      "IsNullable"    : true,
      "ColMaxWidth"   : 40
     },
     {"ColumnNum"     : 2,
      "ColumnName"    : "SYS_NAME",
      "CType"         : 1,
      "CTypeSize"     : 1025,
      "AblType"       : "CHARACTER",
      "NumDecimals"   : 0,
      "IsNullable"    : false,
      "ColMaxWidth"   : 255
     },
     {"ColumnNum"     : 3,
      "ColumnName"    : "ANNUALREVENUE",
      "CType"         : 8,
      "CTypeSize"     : 8,
      "AblType"       : "DECIMAL",
      "NumDecimals"   : 0,
      "IsNullable"    : true,
      "ColMaxWidth"   : 53}
   /* etc */
   ],
  "resultset"    : [
     {"ACCOUNTNUMBER"       : "CC978213",
      "SYS_NAME"            : "GenePoint",
      "ANNUALREVENUE"       : 30000000.0,
      "NUMBEROFEMPLOYEES"   : 265,
      "DESCRIPTION"         : "Genomics company engaged in mapping
                              and sequencing of the human genome
                              and developing gene-based drugs",
      "SLAEXPIRATIONDATE"   : "2013-01-14"}
    /* etc */
    ]
}
```

Other operations

Getting datasource-supported data types

There are multiple SQL data types that are supported by D2C and Easyl. This sample has only been tested against those used by Salesforce.com and basic OpenEdge types as part of the Easyl certification. Since we use SQL Concise types on the ODBC interface, many other data sources should be supported too.

To see what types are supported by your datasource, call the **`GetTypeInfo()`** method on the **`OpenEdge.Data.ODBC.SqlGetTypeInfo`** class. See example below

```
define variable oResultArray as JsonArray no-undo.

oResultArray = oD2CServer:GetTypeInfo().
if valid-object(oResultArray) then
   oResultArray:WriteFile('temp/type-info-all.json', true).
```

Method Name	Return Type	Access	Parameters	Comments
GetTypeInfo	`Progress.Json.ObjectModel.JsonArray`	public		Returns an array of JSON Objects describing the data types supported by the data source.
GetTypeInfo	`Progress.Json.ObjectModel.JsonArray`	public	input `OpenEdge.Data.ODBC.SqlTypeEnum`	Returns a filtered array of JSON Objects describing the data types supported by the data source.

Getting schema information

The `ODBCConnection` object also provides an API for querying the data source about the tables and fields (columns) it contains.

Method Name	Return Type	Access	Parameters	Comments
GetTables	`Progress.Json.ObjectModel.JsonObject`	public	input plChildSchema as logical	Returns schema information for all tables in the data source. If the parameter is true, all child schema elements (initially only columns) are also returned
GetColumns	`Progress.Json.ObjectModel.JsonObject`	public	input pcTableName as character	Returns schema information about all columns for the specified table.
GetTableSchema	`Progress.Json.ObjectModel.JsonObject`	public	input pcTableName as character	Returns table and column schema information for the specified table. Differs from `GetColumns` in that table schema is also included

A truncated example of the output produced by `GetTableSchema` for the Salesforce USER table is below.

Table and column schema

An example of table and column schema is shown below.

```
{"query"   : "SQLTables",
 "tables"  : [
  {"TABLE_CAT"   : null,
   "TABLE_SCHEM" : "SFORCE",
   "TABLE_NAME"  : "USER",
   "TABLE_TYPE"  : "TABLE",
   "REMARKS"     : null,
   "columns"     : [
    { "TABLE_CAT"          : null,
      "TABLE_SCHEM"        : "SFORCE",
      "TABLE_NAME"         : "USER",
      "COLUMN_NAME"        : "ROWID",
      "DATA_TYPE"          : -9,
      "TYPE_NAME"          : "ID",
      "COLUMN_SIZE"        : 18,
      "BUFFER_LENGTH"      : 54,
      "DECIMAL_DIGITS"     : null,
      "NUM_PREC_RADIX"     : null,
      "NULLABLE"           : 0,
      "REMARKS"            : null,
      "COLUMN_DEF"         : null,
      "SQL_DATA_TYPE"      : -9,
      "SQL_DATETIME_SUB"   : null,
      "CHAR_OCTET_LENGTH"  : 54,
      "ORDINAL_POSITION"   : 1,
      "IS_NULLABLE"        : "N"  },
    /* etc */
   ]
 }]
}
```

- The `OpenEdge.Data.ODBC.SqlGetTables` class contains a detailed description of the schema for columns in its static constructor.
- The `OpenEdge.Data.ODBC.SqlGetColumns` class contains a detailed description of the schema for tables in its static constructor.

NOTE: Easyl provides schema mapping from data sources as part of data set construction and reporting capabilities. Therefore, you work with schema from Easyl data sources to construct data sets that are then accessible from the Easyl client. Therefore, when you use SQLGetTables and SQLGetColumns, etc. from the Easyl client driver, you are querying business objects in the configured data sets that map attributes of its business objects. This is indirect access through data set mapping to the underlying columns and tables of the associated data sources which are not directly reference-able through the Easyl driver.

Conclusion

This document describes how to connect your OpenEdge application to the DataDirect Cloud service or Easyl platform - and thus provides the ability to gain access to a variety of Cloud-based data sources - using a simple ABL API. This functionality allows you to use data sources from the DataDirect Cloud or Easyl in order to enrich your OpenEdge application.

The associated project code is also available on Progress Communities, on the Integrating DataDirect Cloud and Progress Easyl into OpenEdge using the ODBC Bridge Sample Applications page. Please feel free to provide comments and feedback, and questions you might have to that page and its parent community.