DataLoad – automating data loading into Oracle Applications

Renier Figuracion BCom, BSc

Systems Administrator/Accountant

University of Canterbury

Introduction

Oracle provides open interfaces into the Applications for automating mass loading of data. However, these interfaces are not available in all areas and a certain amount of skilled work is required to build and use them.

DataLoad is a non-Oracle program developed to allow end users to automate the manipulation of the Application's forms, providing a method of loading data that ensures Oracle's validation code is still applied. All data is entered through Oracle forms so there are no support implications by using this tool and data is validated by the application in the normal way.

Although DataLoad was aimed at Oracle Forms based applications, it can be used with any program running in Windows environment, including Java based software.

DataLoad is available in DataLoad Classic, the freeware version of the program, and DataLoad Professional. A DataLoad Professional licence is approx £250.

This is a non-technical presentation that demonstrates on how to setup and use the program on Oracle Applications 11i environment.

How does DataLoad work?

DataLoad was originally written as an Excel macro, but was rewritten as a standalone Windows application to improve performance, reliability, and functionality.

The program works by replicating user's actions entering data into form fields such as pressing keys, sending keystroke combinations, accessing menu commands, button presses, or mouse clicks.

DataLoad provides two methods in entering data through forms, namely:

- Cut & Paste, and
- Forms Playback.

Cut & Paste

Uses Windows copy & paste technology for faster loading, but it can also simulate keystrokes to enter data to the form fields. When loading data, it calls the low-level Windows code thus the target Application cannot differentiate between data entered by the user at the keyboard or DataLoad. The program processes the loading from the client machine therefore performance is very dependent on the speed of the desktop machine and network. Building the loading template is a little bit cumbersome and not very intuitive. However, this method has far more flexibility when used with other non-Oracle applications.

Forms Playback

Works very similar to Excel macro. When you run Forms in record mode, a file is written detailing everything you did in the forms session. Run forms in playback mode and the forms will do whatever is specified in the file being played back. This method runs many times faster than Cut & Paste because the loading process is generated from the form server. Creating a loading template is very simple and setup is minimal.

Practical uses in our environment

Two years ago, at the University of Canterbury, we had a fresh installation of Oracle Application Release 11i. Previously we were on Release 10.7, but because we re-engineered most of our business process, we decided to configure the system from scratch.

At that time we decided to use DataLoad 3.x to upload our new sets of accounts and setting up all University users. We also initially used it to load our daily exchange rates but later on decided to have a customized program mainly due to user preference.

We managed to integrate Oracle Applications with one of the University's program called Beims, which is used for our works and facilities management. Oracle Accounts Payable and Purchasing data are uploaded to Beims regularly using DataLoad to keep the two systems in sync.

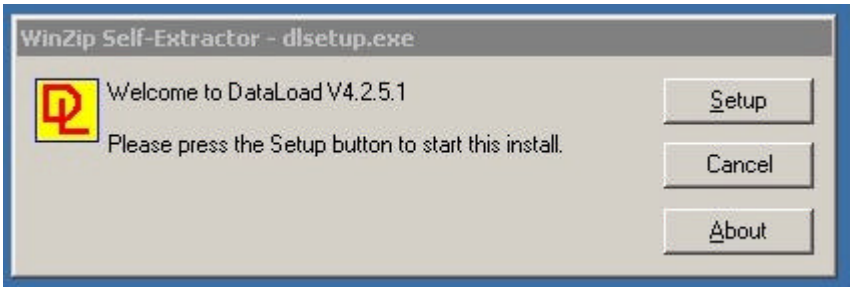
Until three months ago, we were not keen on the practical uses of the program. This was because the old version was a bit cumbersome to build and very dependent on the performance of the client machine and the network. However, DataLoad 4.x is a vast improvement from the previous version. The Forms Playback function alone made us seriously rethink on how we can use the product to gain efficiencies in some areas of our operation. Its seamless integration with Oracle Release 11i makes developing automated data upload easier and faster to deploy.

How to setup DataLoad?

DataLoad version 4.x has been successfully tested with Oracle Applications Release 10.7.0 NCA (post RTU only), 11.0.X & 11i, and on all 32-bit Windows platforms. When using the Forms Playback, you need to be using either Forms 4.5 or Forms 6.x and an Internet Browser supported by Oracle. This presentation is about configuring the system to run DataLoad on Oracle Applications Release 11i environment that uses Forms 6.x.

First thing to do is download the latest version at http://www.dataload.net/downloadV4.htm and install it in your client machine.

The file downloaded is an executable file. When you click on the 'Setup' button, it will automatically install DataLoad, which includes the DataLoad Classic executable, example spreadsheets, and an HTML user guide.



No further configuration is required if you are only intending to use the *Cut & Paste* function. However, to use the *Forms Playback* function, you have to configure Forms 6.x to enable forms recording and playback. Although the tasks are simple, it is strongly recommended that you ask your Systems Administrator or DBA to access and edit these files for you.

The following tasks must be completed to enable the application to use the *Forms Playback* function:

(1) Edit appsbase.htm

This file is typically located in both $OA_HTML\US and $FND_TOP\html\US directory. Ensure both copies are edited and are in the same version.

Step 1 -- Open this file in your preferred html editor and find the following lines:

```
// Special Function Parameters
// ---------------------------
   var xrecord       = "%record%"
```

Now add a variable for the play parameter immediately below the xrecord variable. This should be as follows:

```
var xplay         = "%play%"
```

Step 2 -- Under 'IE Section' section of this file, find the following line;

```
IEhtml += '<' + 'PARAM name="serverArgs" value="module=' + xmodule +
' userid='+ xuserid + ' fndnam=' + xfndnam + ' record=' +xrecord;
```

If the code `"+ ' record=' +xrecord"` is included in this line then it should be removed.

You then have to insert the following lines;

```
if (xrecord != "") {
   IEhtml += ' record=' + xrecord ;
 }
 if (xplay != "") {
   IEhtml += ' play=' + xplay ;
```

The modified portion of the code will appear like this;

```
IEhtml += '<' + 'PARAM name=serverArgs   value="module=' + xmodule
+ ' userid=' + xuserid + ' fndnam=' + xfndnam ;
 if (xrecord != "") {
   IEhtml += ' record=' + xrecord ;
 }
 if (xplay != "") {
   IEhtml += ' play=' + xplay ;
 }
 if (xlog != "") {
   IEhtml += ' log=' + xlog;
```

The changes done on Step 2 only affect Internet Explorer users. For Netscape and Macintosh users, you have to modify respective sections of the code, which can be identified by the use of the variables NShtml and MAChtml.

(2) Edit appsweb.cfg

This file is typically located in both $OA_HTML\US and $FND_TOP\html\US directory. Ensure both copies are edited and are in the same version.

Open this file in a text editor and find the section called 'Special Functionality Parameters'. Look for the 'record' parameter and directly underneath it add the 'play' parameter. It should appear as follows;

```
; 7) Special Functionality Parameters
; -----------------------------------
; Record parameter values include:
; - performance : records server events timings into log file
; - collect     : records Runtime Diagnostic data into log file
; - all         : records Diagnostic and Performance data
; - names       : adds UI names to messeges, no log generated
; - pecs        : old performance data saved into log file
; Otherwise no recording takes place.
record=
play=
;
```

How to create a DataLoad template?

Before you start uploading data to a form, you must first create a DataLoad template that will precisely populate the fields in the target form. Once you have the template, you can paste data on it for uploading.

Cut & Paste

To create a template for the *Cut & Paste* method, read the free article DataLoad for Dummies on the DataLaod website http://www.dataload.net . This article gives you a detailed example on how to create one. The template file will have a .dld extension.

Forms Playback

Creating templates for the *Forms Playback* method is very simple. It is similar to creating a macro by first recording the data entry actions and then playing it back.

- To start a recording session, you have to login to the Application using the URL address format:

  http://host.domain:port/dev60cgi/f60cgi?record=[*filename.fld*]

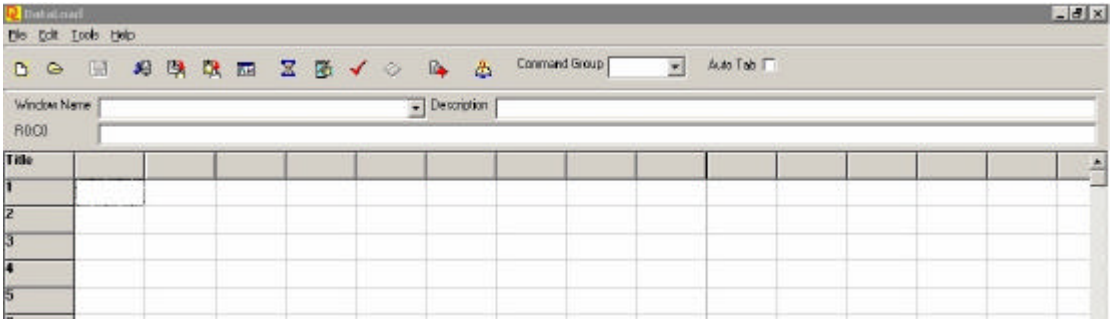
- To start a play session, the URL address format to use is;

  http://host.domain:port/dev60cgi/f60cgi?play=[*filename.fld*]&record=[*log filename.fld*]&code=oracle.forms.engine.MainRT

How to start DataLoad?

Cut & Paste Function

1) When using this function, just double click the program execution file called DataLoad.exe to start DataLoad.



2) Open the DataLoad file (.dld extension) you want to load by clicking **File -> Open** in the toolbar.

   If the file is in Excel or delimited format, you must first import the file to the DataLoad sheet. For files with delimited format, click **File -> Import Data** in the toolbar.

   As for Excel files, you must first open the Excel file and highlight the data you want to import. Then, click the '**Import Data from Excel**' button in the toolbar.

3) Open the target form you want to send the data and make sure the cursor is in the first field.

4) In the DataLoad sheet, go to the 'Command Group' field and pick the Application environment. Then, in the 'Window Name' field, pick the name of the Application form you'll be sending the data to.

5) Click the '**Send data to form**' button to start the upload.

Forms Playback Function

There are two ways to start this function.

- Internet Browser
- Oracle Personal Home Page

Internet Browser

This method of access is done by changing the URL that launches the forms session to include the parameters which control the record and playback. There are three parameters that must be used:

- record = [*filename*] Filename is the full path and file name where the forms session should be recorded to.
- play = [*filename*] Filename is the full path and file name which will be used to control a forms playback session.
- code = oracle.forms.engine.MainRT – This ensures the forms are drawn during playback sessions.

For example, to record a Forms Playback file to 'tmp/record.fld' your URL address will look like this;

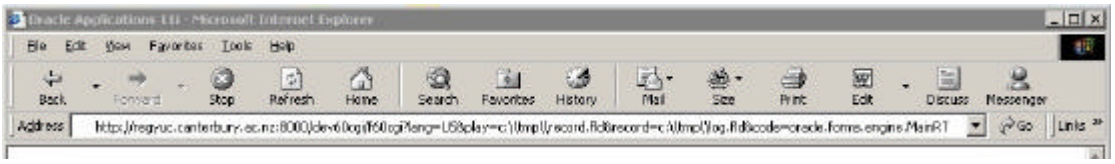
http://host.domain:port/dev60cgi/f60cgi?record=/tmp/record.fld

To playback above record file and write a log file to 'tmp/log.fld', change the URL to as follows;

http://host.domain:port/dev60cgi/f60cgi?play=/tmp/record.fld&record=/tmp/log.fld&code=oracle.forms.engine.MainRT

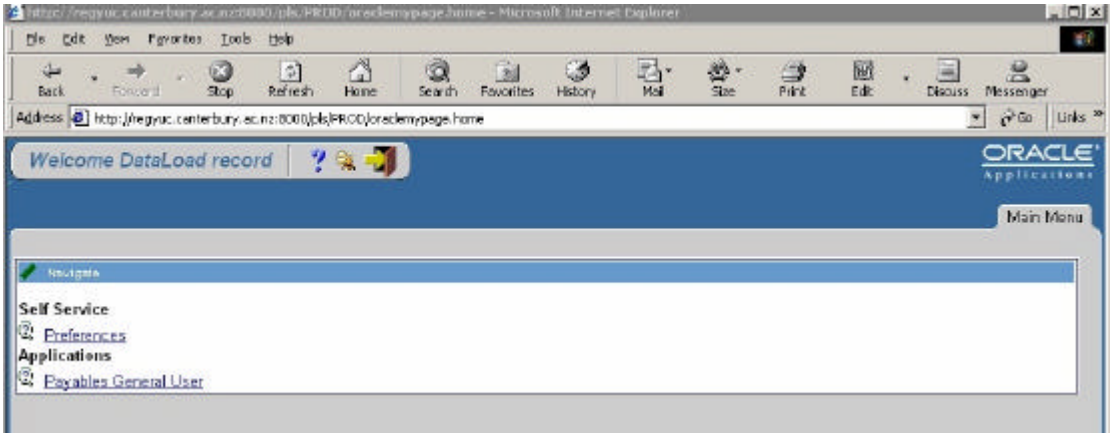
Unix environment:

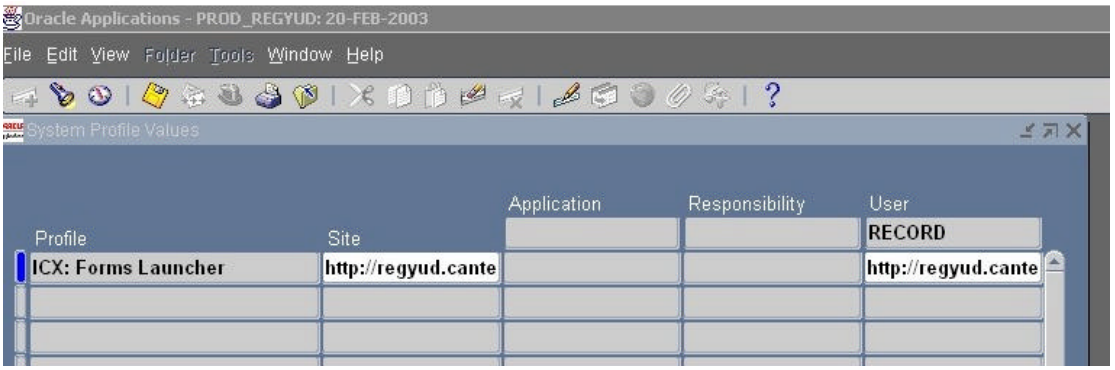
Windows environment:



Oracle Personal Homepage

This method requires setting the Application profile option called 'ICX: Forms launcher'. You can either set this profile at Site, Application, Responsibility, or User (highly recommended) level depending on how you want to control the recording or playing session. Enter the URL address shown above as the ICX: Forms launcher profile value. This will automatically enable the recording or playing session when you login to the Application. For example, if you set the profile at User level, the recording or playing session will immediately start when the User login to the system.





How to get support?

DataLoad Professional version provides support but the freeware version, DataLoad Classic, only have very limited support. They have a free email support (support@dataload.net) to users who have difficulty downloading, installing or starting the program.

However, there are other resources you can freely use if you run into problems, namely:

- DataLoad documentation – http://www.dataload.net/help/
- DataLoad FAQ – http://www.dataload.net/help/faq.htm
- DataLoad Forum – http://www.dataload.net/forums/DataLoad/index.html

Conclusion

There are many ways to automate loading of data into Oracle Applications and using DataLoad is one of them. It uses the Application's built-in forms to load data just like a user enters data into the system, thus ensuring proper data validation. Because it does not use conventional Application interface tables, nor does it populate directly database tables, Application Developers specialist skills are not required. This itself offers a major cost savings in any project.

Furthermore, no programming scripts are required to develop the loading template. Similar to creating macros, it is as simple as recording data entry actions and playing it back to upload data. This means deployment of the program to the Production environment is much faster than conventional way developing interface programs.

DataLoad also simplifies integrating non-Oracle sub-system with the main Oracle Application system. As long as the sub-system is a Window based product and supports cut and paste, data migration between systems can be done with minimal technical expertise. This is possible because by loading data from the forms level, knowledge on the underlying complexities of the programs are not required.

About the Author

Renier worked in Papua New Guinea for twelve years as an Accountant, before upgrading his skills by graduating with an Information Technology degree from the University of Canterbury. Renier started work at the University of Canterbury as Systems Administrator/Accountant in 1997, where he is involved in administering and developing Oracle Applications from Release 10.4 through to 11.5.7. Renier took a significant role in the successful re-implementation of release 11i, including implementing new modules, and developing interfaces and customisations to improve the efficiency and use of the systems.