



## Alternative Approaches to Data Conversion

Jay Pradhan 8/17/2012

NCOAUG Summer Training 2012

**ORACLE®**

Platinum  
Partner

# Statement of Confidentiality & Disclaimers

Terillium is pleased to present the attached Presentation for the software and/or services referenced therein. The attached Presentation and its contents (collectively, "Presentation") are the confidential and proprietary information of Terillium and Oracle. The Presentation may be used by You solely for evaluation of a business relationship between Terillium, Oracle and You.

The Presentation was derived from material produced by both Oracle and Terillium and the contents hereof are subject to change without notice. The information contained within is not a guarantee of services and no contractual obligations are formed either directly or indirectly by this Presentation. This Presentation may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

The Presentation may include selected third party data, information, research and/or reference materials (collectively, "Third Party Information"). Terillium does not warrant the accuracy or endorsement of the Third Party Information, which is provided "as-is."

Further, this Presentation is not a commitment to deliver any future material, code, or functionality. The development, release, and timing of any features or functionality described in this Presentation remains at the sole discretion of Terillium and Oracle.

The Presentation cannot account for all risks and other factors that may affect results or performance, or for changes in your business practices or operating procedures that may be required to realize results or performance, that are projected or implied in the Presentation.

The Presentation, and any discussion or negotiation of The Presentation between Terillium, Oracle and You, is for informational purposes only and is not an offer by, commitment from, or contract with Terillium to provide any software or services. The Presentation is subject to change at Terillium's sole discretion.

# Agenda

- Introduction : What is data conversion?
- What challenges do data conversion efforts offer?
- What are the traditional methods of data conversion?
- What are the pros and cons of traditional data conversion?
- What does dataloader do?
- What does dataloader professional do?

# Agenda

- How it works?
- What are the differences in approach between forms playback and traditional API conversion.
- What advantages do Dataload forms playback offer.
- What are the limitations of Dataload forms playback face?
- Best practice recommendations.
- Gotchas

# Introduction: What is Data Conversion

- ✓ One time effort to move data from legacy to new system.
- ✓ Can include non transactional data such as items, BOM, Routings, Customers, Suppliers, Pricelists and Installed Base.
- ✓ Can include transaction data such as open Sales orders, Invoices, POs, Quotes, RMAs and Service Requests.
- ✓ Inevitable and technically most challenging task of new implementation projects.

# Traditional methods of Data Conversion

- ✓ Flat files extracted in excel or comma delimited format from legacy system.
- ✓ Files loaded into staging table in Oracles. Sometimes into interface tables.
- ✓ Data imported using seeded import program. Example : Sales orders, Invoices.
- ✓ Data loaded in system with public APIs written by Oracle.

# Challenges In Traditional Methods

- ✓ Trying to fit square peg in a round hole. Legacy data and eBusiness Suites data being structurally different.
- ✓ In-coherent, inaccurate and duplicate values in legacy system.
- ✓ Different ways of validation in legacy compared to EBS.
- ✓ Too many links in the chain of communication : Legacy programmer, legacy BA, EBS programmer and EBS BA all have to be on the same page.
- ✓ Additional burden on the programmers in EBS to handle fall-outs, report them to business analysts on both sides and bring everyone on the same page.

# What does Dataload Pro do?

- ✓ Macro Loads – a common feature in the free version of Dataload.
- ✓ Load data on using browser based self service forms (HTML / JSP forms).
- ✓ Load data using forms playback feature.



# Forms Playback Feature: How it works?

- ✓ Two logins in EBS : One for recording keystrokes and another for playing back the recording.
- ✓ Login with recording ID and enter two transactions in forms.
- ✓ The transactions must be entered in identical manner.
- ✓ Dataload Pro records the transactions and creates an FLD file on application server with 2 rows representing the two records that were created.
- ✓ Copy-paste data extract from legacy into the file and place it back on application server.
- ✓ Login with playback ID and open the same form. As soon as form launches, the playback file is played and records are loaded thru forms direction from application server.

# Difference between Forms Playback and Macro load in Dataload

- ✓ In Macro load, the data is loaded by passing keystrokes to Oracle forms on an active session in the user's machine. In Forms playback, the playback file is on the application server and data is loaded from application server to the database directly.
- ✓ Forms playback is about 10-15 times faster than Macro loads.
- ✓ Forms playback does not depend on VPN connections or network speeds as it only relies on connection between application server and the database.

# Forms Playback : One-time setups in EBS and application server

- ✓ Create two user accounts in EBS : One will be used for recording manual keyed entry of two records. Another will be used to playback the recording and load data.
- ✓ Grant responsibility to both IDs. All responsibilities that have the forms to where data needs to be converted.
- ✓ Setup profile values 'ICX: Forms launcher' for user level for both new user accounts.
- ✓ Place a noplay.fld dummy file in Forms Trace Directory. The directory would be the one that's listed in \$FORMS\_TRACE\_DIR environment variable.

# Forms Playback : Typical process of loading data.

- ✓ User logs in the system with recording ID and enter 2 transactions back to back and saves them and logs out.
- ✓ The system writes a .fld file in the application server in directory setup in forms trace files. Copy this .fld file and bring it to the local laptop/desktop from application server.
- ✓ Open the file in Dataload professional and edit the file to add hundreds/thousands of records by replacing the 2 that are already in there. They must be the same format. Save the .fld file.
- ✓ Place the file back into forms trace directory and then login into EBS.
- ✓ Launch the form from which the two initial records were inserted. The playback starts and data is loaded.

# Key Features of Forms Playback.

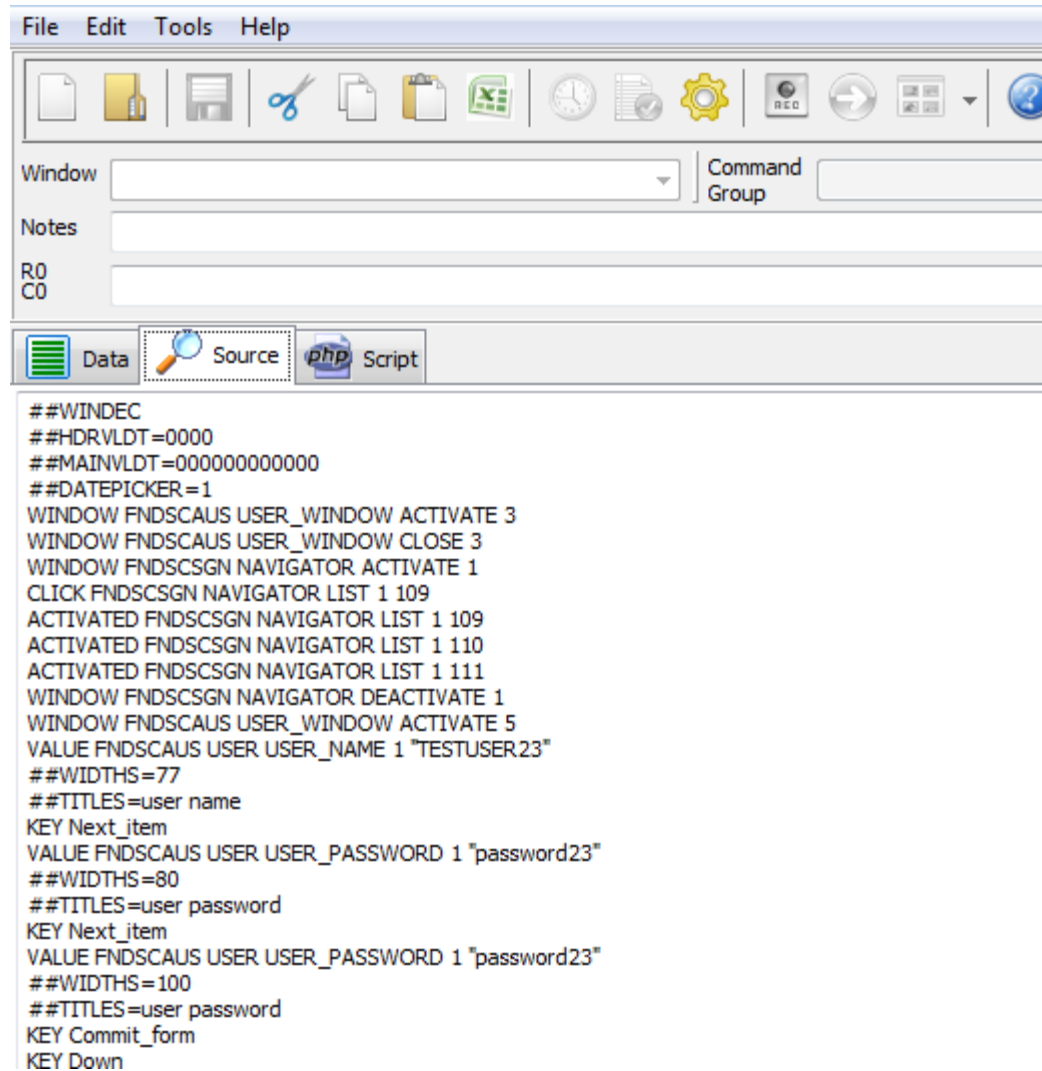
- ✓ It runs from application server unlike the macro based loads of Dataload.
- ✓ Much faster than macro loads since its loading data directly from application server and therefore the network speeds and VPN stability is immaterial.
- ✓ User can continue working while the data is being loaded in the Oracle forms window. In Macro loads, the machine cannot be used for anything else while the load is running.
- ✓ Dataload.com has a screen cast video that shows what forms playback looks like.  
<http://www.dataload.com/video/users/users.html>

# Example of a .FLD file - 1.

The screenshot shows a software application window with a menu bar (File, Edit, Tools, Help) and a toolbar. Below the toolbar are fields for 'Window', 'Command Group', 'Notes', 'R0', and 'C0'. On the right, there is a 'DataLoad Quick Links' section with a 'DL' logo. Below these elements are tabs for 'Data', 'Source', and 'Script'. The main area displays a table with the following data:

Main Load	user name	user password	user password
1	TESTUSER23	password23	password23
2	TESTUSER24	password24	password24
3	TESTUSER25	password25	password25
4	TESTUSER26	password26	password26
5	TESTUSER27	password27	password27
6			
7			
8			
9			
10			
11			
12			

## Example of a .FLD file - 2.



The screenshot shows a software window with a menu bar (File, Edit, Tools, Help) and a toolbar with various icons. Below the toolbar are fields for 'Window', 'Command Group', 'Notes', 'R0', and 'C0'. At the bottom, there are three tabs: 'Data', 'Source', and 'Script'. The 'Script' tab is active, displaying the following text:

```
##WINDEC
##HDRVLDT=0000
##MAINVLDT=000000000000
##DATEPICKER=1
WINDOW FNDSCAUS USER_WINDOW ACTIVATE 3
WINDOW FNDSCAUS USER_WINDOW CLOSE 3
WINDOW FNDSCSGN NAVIGATOR ACTIVATE 1
CLICK FNDSCSGN NAVIGATOR LIST 1 109
ACTIVATED FNDSCSGN NAVIGATOR LIST 1 109
ACTIVATED FNDSCSGN NAVIGATOR LIST 1 110
ACTIVATED FNDSCSGN NAVIGATOR LIST 1 111
WINDOW FNDSCSGN NAVIGATOR DEACTIVATE 1
WINDOW FNDSCAUS USER_WINDOW ACTIVATE 5
VALUE FNDSCAUS USER USER_NAME 1 "TESTUSER23"
##WIDTHS=77
##TITLES=user name
KEY Next_item
VALUE FNDSCAUS USER USER_PASSWORD 1 "password23"
##WIDTHS=80
##TITLES=user password
KEY Next_item
VALUE FNDSCAUS USER USER_PASSWORD 1 "password23"
##WIDTHS=100
##TITLES=user password
KEY Commit_form
KEY Down
```

# Technical Differences in Approach : Traditional Versus Dataload Forms Playback

- ✓ In Forms playback, the load is done by Oracle form from Application server. In traditional API driven approach, data is loaded either thru a seeded import program or thru wrapper containing public APIs.
- ✓ Much faster than macro loads since its loading data directly from application server and therefore the network speeds and VPN stability is immaterial.
- ✓ User can continue working while the data is being loaded in the Oracle forms window. In Macro loads, the machine cannot be used for anything else while the load is running.



# Advantages of Forms Playback

- ✓ No programming skills needed. Business analysts or functional users can load data.
- ✓ If the client users are up for it, they can also use Dataload pro to load data into EBS.
- ✓ Forms playback technique is much faster than the usual Macro Loads in Dataload. Approximately 1000-2000 records can be loaded per hour.
- ✓ Load stops as soon as there is an error. This forces validation to occur on the front end before bad data is loaded or an exception file is generated.
- ✓ Data that goes in the system is validated and vetted by Oracle forms. There is never a chance of data being loaded that will not pass forms validation – as it happens occasionally with APIs.

# Advantages of Forms Playback

- ✓ Lookups, value-sets and other functional setups are corrected and improved as and when data is loaded, as opposed to going thru the fall-out and exceptions files and then fixing them in the system.
- ✓ If the functional users or business analysts are loading data into EBS, then there is less chance of loading technically correct but functionally wrong data, since they are familiar with business impact of data being loaded.
- ✓ Fewer chances of communication gaps.
- ✓ Take advantage of periods of downtime for functional consultants if available – usually between CRP1 and CRP2 on small to mid-size implementations. This may significantly reduce data conversion budgets and improve utilization of functional consultants.

# Limitations of Forms Playback

- ✓ Slower than API driven programmatic loads. It can load 1000-2000 records per hour – depending on the number of items in record.
- ✓ Requires monitoring. Unlike programmatic loads, Dataload requires someone to watch the load and intervene when it stops due to an error. Programmatic loads driven by import programs or APIs will dump exceptions and fall-outs into a file to be reviewed later.
- ✓ If data loaded is not acceptable, either the last backup has to be restored or technical fix may be needed to truncate data. This may be a limitation of API load too. The functional users may not be able to remove loaded data and so restore from last back-up is probably the only alternative.

# Best Practice Recommendations

- ✓ **Determine Scope:** Determine if there are data conversions that are relatively small to mid-size. Dataload forms playback is ideally suited for less than 20,000 records per conversion. If you have 30,000 rows in your conversion, its better suited for API load. Retain the usual API methodology for data conversions that have over 20,000 rows and try remaining loads with forms playback.
- ✓ **Start Simple :** Start with simpler conversions. User accounts may be simple with 2-3 items per row. Then move on to more complex conversions that have 20-30 items per row – such as item master.
- ✓ **Validate Data :**Check to quality of data before starting loads. This may help prevent the frequent interruptions whenever there is bad data that forms wont accept. This can be done in excel by comparing values in data to values in lookups and value-sets.

# Gotchas

- ✓ Functional users who need to run Dataload Pro have to be comfortable moving files back and forth between local machine and application server. It can be made very easy with tools like Putty or WinSCP – available as free-ware.
- ✓ Recording the first few conversions can be frustrating. You have to enter two records in identical manner, keystroke by keystroke.
- ✓ Running forms playback require constant monitoring as opposed to import program or API wrappers called by concurrent programs.
- ✓ Setups have to be complete ahead of time on which data conversions are dependent. This can be an advantage in the long run but it front-loads the fall-out issues tied to setups.

