

Forms Playback: Unlocking Oracle's Hidden Tool for Fast Data Entry



Executive Overview

Forms Playback is an undocumented feature of Oracle Applications that can be used to speed up data entry of large amounts of data by up to 10 times that of conventional methods. It is an invaluable tool during the time of implementations, conversions or routine data maintenance activities. Its many benefits include:

- ✓ 10 times faster than Data Loader
- ✓ Greater accuracy and reliability
- ✓ Ability to run in the background while working on other windows
- ✓ Recording a log of all actions for review
- ✓ Option to run in "silent" mode which is even faster
- ✓ Cost-effective – no additional licensing costs or programming efforts

This innovative solution of using Forms Playback ended up saving thousands of hours of Business effort at a US Fortune-50 retailer. It was appreciated by Business and IT Management alike as an innovative solution to a ubiquitous problem.

TABLE OF CONTENTS

EXECUTIVE OVERVIEW 2

PURPOSE OF THE DOCUMENT 4

WHY FORMS PLAYBACK?..... 4

BACKGROUND 4

OVERVIEW..... 5

SETUP 6

USAGE EXPLAINED WITH AN EXAMPLE..... 7

CONCLUSION 9

Purpose of the document

The scope of this document is to explain the process of using Forms Playback functionality to achieve fast data load speeds along with greater accuracy.

Why Forms Playback?

Every Oracle Applications implementation requires setups and configuration information to be loaded before the system goes live. Furthermore, there are often requirements to load or amend data in the system once it has gone live as part of conversion or periodic activities.

Oracle provides open interfaces that are normally used for this purpose, however these interfaces are not available in all areas and programming skills are required to use them.

The other non-technical method is to use Data Loader to allow end users to automate the data entry. 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. However this has the disadvantage of being slow and error-prone and requires constant monitoring.

The answer to these problems is Forms Playback – this is a special undocumented feature of Oracle Applications that allows fast and accurate automation of data-entry into the application. Also, since it uses the front-end forms it goes through all the front-end validations. It can speed data loading by up to ten times. It also records a log of all its actions for audit or support purposes.

Background

Oracle Forms can record all the events in a forms session to a text file, which can then be replayed by the forms software at a later date. By editing the file produced in the manner described here, Oracle Forms will replay your data and make your changes or data loads for you.

Before this functionality can be used Forms software must be enabled to run in this mode. We achieved this by editing the URL which starts Oracle Forms. Furthermore, one URL will be required for recording and another for playback. We involvement our DBA team for this step and it is fully described in the Setup section.

Before recording the Oracle Forms session we planned exactly what we were going to do. Since our recording would be replayed many times over, any mistakes would also be replicated for each row of playback later.

Once our load was built it had to be tested. This involved copying the file produced by Data Load to a location accessible to our Forms software and running Oracle Forms using the playback parameter. This is explained with the help of an example.

Overview

The following is an overview of the steps we followed in a typical Forms Recording and Playback session:

1. We performed the Forms Playback setup as described in the Setup section which follows.
2. We recorded a load using the recording URL obtained from step 1. For ease of editing, we made sure that we performed similar updates and recorded exactly two of these, thus helping us to locate the fields that stayed the same from those that changed.
3. We copied the load file recoded in step 2 from the Forms server to our PC.
4. We opened the file in a text-editor and identified the **header** (the portion that appears just once at the very start), the repeating rows or **body** (the actual data entry) and the **footer** at the bottom (optional and used to close the window).
5. We replaced and added to the body the actual data that needs to be loaded.
6. We saved our changes and copied the file back to the server, ensuring the file was named and located as detailed in the Setup section.
7. We started the playback by loading the playback URL in the browser.

All the above steps are also explained with the help of an example, where we used Excel to automatically update 1,000 GL accounts.

Setup

1. We open an Oracle Apps session, right-clicked on a blank space in the loading window and selected Properties to copy the Address (URL):



2. We broke the URL of the Oracle window into 4 components as shown below. The URL:

```
http://app-
fin1dev1.server.com:11160/dev60cgi/f60cgi?colorScheme=BLUE&lang=US&env=NLS_LANG='AMERICAN_AM
ERICA.UTF8'+FORMS60_USER_DATE_FORMAT='DD-MON-
RRRR'+FORMS60_USER_DATETIME_FORMAT='DD-MON-
RRRR%20HH24%3AMI%3ASS'+NLS_DATE_LANGUAGE='AMERICAN'+NLS_SORT='BINARY'&form_params=
+config='FIN1DEV1'+icx_ticket='.876170282'+resp=SQLGL%2FABSN_GL_SUPER_USER'+secgrp='STANDAR
D'&encoding=UTF-8
```

was broken into:

- 1) http://app-fin1dev1.server.com:11160/dev60cgi/f60cgi?
- 2) colorScheme=BLUE&lang=US&env=NLS_LANG='AMERICAN_AMERICA.UTF8'+FORMS60_USER_DATE_FORMAT='DD-MON-RRRR'+FORMS60_USER_DATETIME_FORMAT='DD-MON-RRRR%20HH24%3AMI%3ASS'+NLS_DATE_LANGUAGE='AMERICAN'+NLS_SORT='BINARY'
- 3) &form_params=+config='FIN1DEV1'+icx_ticket='.876170282'+resp=SQLGL%2FABSN_GL_SUPER_USER'+secgrp='STANDARD'
- 4) &encoding=UTF-8

3. We removed the 3rd component to make it look like this:

```
http://app-
fin1dev1.server.com:11160/dev60cgi/f60cgi?colorScheme=BLUE&lang=US&env=NLS_LANG='AMERICAN_AM
ERICA.UTF8'+FORMS60_USER_DATE_FORMAT='DD-MON-
RRRR'+FORMS60_USER_DATETIME_FORMAT='DD-MON-
RRRR%20HH24%3AMI%3ASS'+NLS_DATE_LANGUAGE='AMERICAN'+NLS_SORT='BINARY'&encoding=UTF-
F-8
```

4. We added the below code to it:

- 1) For Recording:
&record=/tmp/dataload/record_file.fld
- 2) For Playback:
&record=/tmp/dataload/reclog.fld%20play=/tmp/dataload/record_file.fld&code=oracle.forms.engine.MainRT

5. This resulted in the following URLs:

- 1) For Recording:

http://app-fin1dev1.server.com:11160/dev60cgi/f60cgi?colorScheme=BLUE&lang=US&env=NLS_LANG='AMERICAN_AMERICA.UTF8'+FORMS60_USER_DATE_FORMAT='DD-MON-RRRR'+FORMS60_USER_DATETIME_FORMAT='DD-MON-RRRR%20HH24%3AMI%3ASS'+NLS_DATE_LANGUAGE='AMERICAN'+NLS_SORT='BINARY'&encoding=UTF-8&record=/tmp/dataload/record_file.fld
- 2) For Playback:

http://app-fin1dev1.server.com:11160/dev60cgi/f60cgi?colorScheme=BLUE&lang=US&env=NLS_LANG='AMERICAN_AMERICA.UTF8'+FORMS60_USER_DATE_FORMAT='DD-MON-RRRR'+FORMS60_USER_DATETIME_FORMAT='DD-MON-RRRR%20HH24%3AMI%3ASS'+NLS_DATE_LANGUAGE='AMERICAN'+NLS_SORT='BINARY'&encoding=UTF-8&record=/tmp/dataload/reclog.fld%20play=/tmp/dataload/record_file.fld&code=oracle.forms.engine.MainRT

6. At the Forms Server, the DBA executed the following steps:

```
cd /tmp
mkdir dataload
cd dataload
touch reclog.fld
touch record_file.fld
chmod 666 *.fld
```

Usage explained with an example

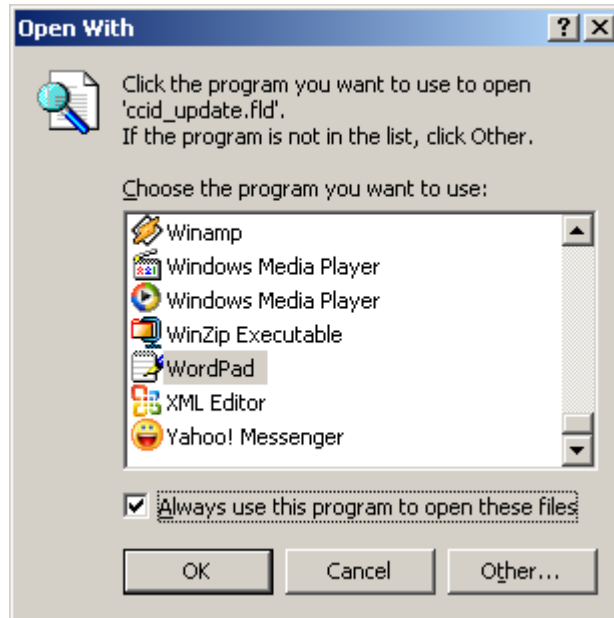
We will explain the usage with the help of an example where we need to disable 1,000 active GL code combinations (CCID s):

1. We associate .FLD file extension with WordPad:
 - a. We opened a file similar to the one below (it is a session recorded with Forms Playback where we have retained just the **header** and stripped the **body** and **footer**):



ccid_update.fld

- b. From the dialog box that came up, we chose *Open With > Choose Program...*
 - c. From the list that came up, we selected WordPad and checked the box *"Always use this program to open these files."*



2. We left this file open. This file would hold the macro commands for the Forms playback functionality. We changed the username, password and responsibility to the appropriate values in lines numbered 2, 4 and 6 and scrolled to the very end.

3. We opened the worksheet CCID_update.xls:



Z:\NewEnterprise\
Databad\CCID_upda

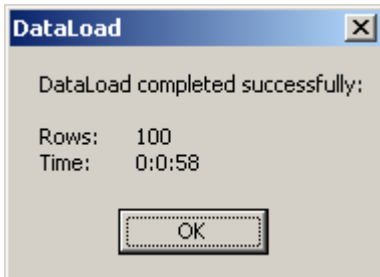
- a. This is a template for updating 1,000 CCIDs at a time. We entered complete code combination strings (1,000 different valid values) in column C of the worksheet.
 - b. We pressed *Alt + Shift + Ctrl + F9* to recalculate the formulas in column A of the worksheet.
 - c. Now column A holds the values for the macro. We copied the entire contents of column A.
 - d. We pasted this at the bottom of the file *ccid_update.fld*.
 - e. Pasting the values from Excel resulted in all the cells being wrapped in "quotes". These are not needed and were deleted by doing a replace all to replace the quotes with a blank value. We made sure that the quotes around the responsibility ("GL Super User") under Line # 6 do not get deleted.
 - f. We saved this as a text file named *record_file.fld*.
4. We asked DBA s to upload the file to the forms server (taking care to upload in ASCII).
5. These files were executed by loading the special URLs described in the Setup section.
6. The log for the session was recorded in the *reclog.fld* file.

Similarly, this method can be used wherever repetitive date entry tasks need to be automated.

Conclusion

Comparison of Standard Data Load and Forms Playback:

Disabling 100 CCID s took **58 – 60 seconds** in Standard Data Load:



Disabling 100 CCID s took **12-15 seconds** using Forms Playback i.e. it was **4-5 times faster** than using Data Loader. The results improve even further with larger loads.

To summarize, forms playback has the following benefits:

- ✓ **Greater speed:**
Since the playback files are on the forms-server there is no latency in sending the data
- ✓ **Greater accuracy and reliability:**
Since Oracle Forms runs the data load instead of a separate tool, it is never out of sync.
- ✓ **Cost-effective:**
It does not require additional licensing or programming efforts since the capability comes in-built with Oracle Applications
- ✓ **Ability to run in the background** while working on other windows unlike conventional Data Load
- ✓ **Recording a log** of all actions for audit or support purposes
- ✓ Option to run in **silent mode** which is faster or normal mode which shows the actions as they happen