

Customization Survival Guide: How to Use E-Business Utilities to Migrate Your Custom Code

Brad Simmons and Donna Campbell
Los Alamos National Laboratory



Los Alamos National Laboratory is a premier national security research institution, delivering scientific and engineering solutions for the nation's most crucial and complex problems. Our primary responsibility is ensuring the safety, security, and reliability of the nation's nuclear deterrent.

The Big Picture

- Configuration Management and challenges for E-Business suite
- Oracle Utilities Can Help (or hurt)
 - FNDLOAD
 - Java XDOLoader
 - adadmin
- Obligatory unix code
- Simple, practical configuration process
- Questions & (maybe) Answers

Configuration Management

- Extensive customization or extension of E-Business suite can cost millions to maintain and manage.
- Many companies begin with a “no customizations” mandate, but often find it impossible to enforce.
- Configuration Management should be able to identify what versions of what software are where.

Oracle Applications “Extras”

- Customizations require “the usual suspects” plus additional “registrations” of objects via the System Administrator responsibility or some equivalent process
 - Register a form
 - Define a value set
 - Define a concurrent program
 - Register an XML Publisher template
 - Etc.

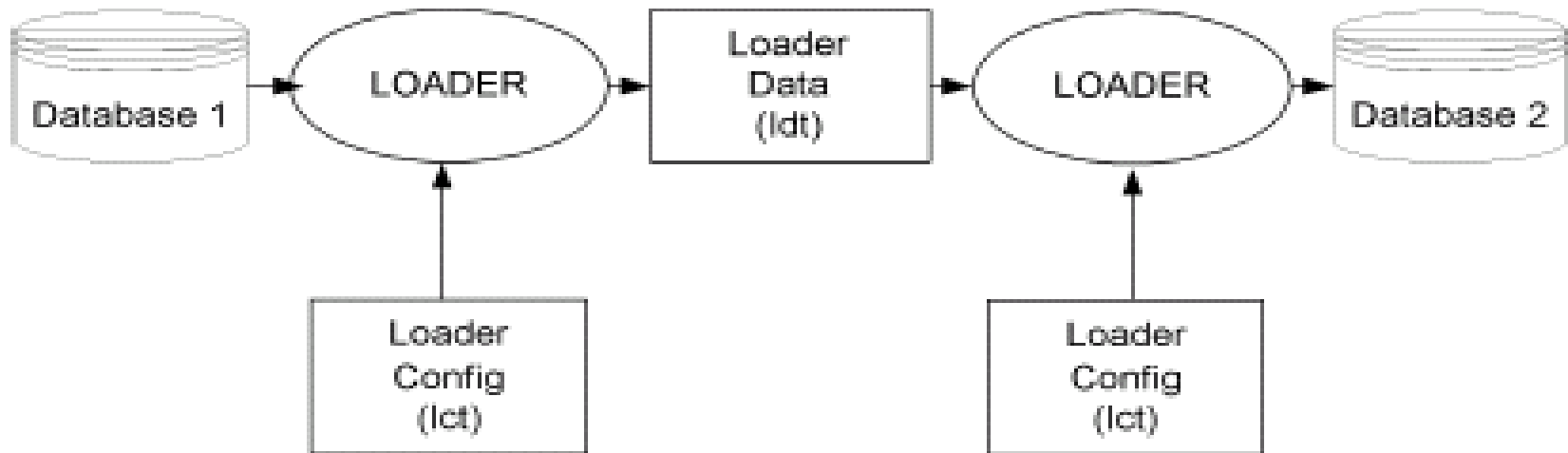
Problem: Manual Apps Registration

- Manually repeating the registration process for each Apps instance is time-consuming and error-prone
- A Configuration Management process needs to manage all pieces of the customization effort – including the “registration” steps.
- Manual or automated, the process needs to be consistent, repeatable, and (usually) auditable.

FNDLOAD: Intro

- The FNDLOAD utility is defined as a “generic loader”
- It can be used to download application data from an Oracle Applications instance into a portable, editable text file.
- The text file can be uploaded into another Oracle Applications instance.

FNDLOAD: High Level Diagram



FNDLOAD: Terminology

- FND Objects (Menu, Responsibility, Concurrent Program, etc.) used to support the operation of the Applications are often named differently but refer to the same concept
 - FND object
 - ‘Seed data type’
 - Entities
 - Objects

FNDLOAD: Modes

- **DOWNLOAD**
 - Data downloaded from the database into a text file
- **UPLOAD**
 - Data uploaded from a text file into the database

FNDLOAD: files used

- Configuration file (.lct)
 - Different lct files for different types of entities
 - Defines the structure of the data
 - Defines the access method for the data
 - Supports master-detail records and foreign key reference relationships
 - Used in both download and upload
 - Readable text files
 - Often contain additional information about parameters used by the file

FNDLOAD: files used (con't)

- Data file (.ldt)
 - Created during a download
 - Used during the upload
 - Somewhat like an XML file because it contains a 'data template' as well as the actual data
 - Structure depends on .lct file, so the creation and use of these files is closely related to the .lct file version

FNDLOAD: command line

- FNDLOAD apps/pwd 0 Y mode configfile datafile entity [parameter....]
 - Mode = UPLOAD | DOWNLOAD
 - Configfile = lct file, depends on entity
 - Datafile = created or used
 - Entity = type of entity
 - Parameter(s) = varies by entity / lct file

FNDLOAD: Entity types = lct file

- Different entities require different lct files
- Located in `$FND_TOP/admin/import` or `$FND_TOP/patch/115/import`
- Different versions delivered with patches will also change resulting datafile
- Menus, Forms, Functions = `afsload.lct`
- Responsibilities = `afscursp.lct`
- Concurrent Programs = `afcpprog.lct`
- Other entity types

FNDLOAD: example commands

- MENU:
 - FNDLOAD apps/apps O Y DOWNLOAD
\$FND_TOP/patch/115/import/afsload.lct menu.lct
MENU MENU_NAME=BOM_NAV
- FORM:
 - FNDLOAD apps/apps O Y DOWNLOAD
\$FND_TOP/patch/115/import/afsload.lct form.lct
FORM FORM_NAME=\${shortname}

FNDLOAD: example commands

- KEYFLEX:

- FNDLOAD apps/\$PASSWD O Y DOWNLOAD
 \$FND_TOP/patch/115/import/affload.lct
 keyflex.ldt KEY_FLEX
 P_LEVEL='COL_ALL:FQL_ALL:SQL_ALL:STR_O
 NE:WFP_ALL:SHA_ALL: CVR_ALL:SEG_ALL'
 APPLICATION_SHORT_NAME=\${appshortname}
 ID_FLEX_CODE=MCAT
 P_STRUCTURE_CODE=\${shortname}

FNDLOAD: <Bang!> <Ouch!>

- Don't shoot yourself in the foot – think about data preservation
- FNDLOAD may insert, update, or even delete rows in the underlying FND tables!
- Rules as outlined in Sys Admin Guide – Configuration
 - owner, last_update_date
- lct file provides insight for behavior

FNDLOAD: Failures

- CHECK log files!
- Different version lct file vs/ ldt file
- No data in ldt file, just template information
- Missing data dependencies
- Other possibilities

FNDLOAD: Great TIP! Copy entities to create new entities

- Download an existing entity
- Created ldt file is editable!
- Change values as needed
- Upload ldt file
- Enjoy new entity!

FNDLOAD: Concurrent Program

- Can use FNDLOAD via concurrent program interface.
- Seeded definition exists, but not usable in original state
- Copy and edit to make usable
- Allows use of pick lists, etc, to facilitate downloads.

FNDLOAD: Concurrent Program

The screenshot shows the Oracle Applications Concurrent Programs configuration window. The main window is titled "Concurrent Programs" and displays details for the "Generic Loader" program with the short name "FNDLOAD".

Program Details:

- Program: **Generic Loader** (Enabled)
- Short Name: **FNDLOAD**
- Application: **Application Object Library**
- Description: **FND Generic Data Loader**

Executable Details:

- Name: **FNDLOAD**
- Method: **Spawned**
- Options: [Empty]
- Priority: [Empty]

Request Details:

- Type: [Empty]
- Incrementor: [Empty]
- MLS Function: [Empty]
- Use in SRS
- Run Alone
- Enable Trace

Concurrent Program Parameters Window:

- Program: **Generic Loader**
- Application: **Application Object Library**
- Conflicts Domain: [Empty]
- Security Class: [Empty]

Seq	Parameter	Description
1	[Empty]	[Empty]
2	[Empty]	[Empty]
3	[Empty]	[Empty]
4	[Empty]	[Empty]

Validation Details:

- Value Set: [Empty]
- Default Type: [Empty]
- Required
- Enable Security

A black oval highlights the "Concurrent Program Parameters" window, specifically the table of parameters and the validation section below it.

FNDLOAD: Concurrent Program

Oracle Applications - 11.5.10.2 Upgrade -- from Apr 20 QYMAN

File Edit View Folder Tools Window Help

Concurrent Programs

Program: LANL FNDLOAD Program for Concurrent Programs Enabled

Short Name: **XXLM_FNDLOAD**

Application: LANL MRP Customizations

Description: Plan Design Copy - Export to File

Executable

Name: **FNDLOAD** Options:

Method: **Spawned** Priority:

Request

Type:

Incrementor:

MLS Function:

Use in SRS

Run Alone

Enable Trace

Copy to...

Concurrent Program Parameters

Program: LANL FNDLOAD Program for Concurrent Programs

Application: LANL MRP Customizations

Conflicts Domain: Security Group:

Seq	Parameter	Description	Enabled
1	Mode	Mode	<input type="checkbox"/>
2	Loader Configuration File	Loader Configuration File	<input type="checkbox"/>
3	Data File	Data File	<input type="checkbox"/>
4	Entity	Type of Object being downloaded/loaded	<input type="checkbox"/>

Validation

Value Set: **240 Characters** Description: **240 Characters**

Default Type: **Constant** Default Value: **/ora_apphome/up**

Required Enable Security Range:

Display

Display Size: **50** Description Size: **50**

Concatenated Description Size: **25** Prompt: **Loader Configura**

Other Oracle Utilities

- XML Publisher data definition and template registration via FNDLOAD, but physical template files require a java tool
 - `java oracle.apps.xdo.oa.util.XDOLoader <parms>`
 `-lct_FILE $XDO_TOP/patch/115/import/xdotmpl.lct`
- OAF pages use class.jar files, but xml pages need to be loaded
 - `java oracle.jrad.tools.xml.importer.XMLImporter`
- Non-interactive AD administration
 - create defaultsfile via interactive session

Utility Unix – entity type

```
## assumes APPS environment is properly set
```

```
##
```

```
## Prompt for type and parameter value
```

```
##
```

```
echo "Enter a category
```

```
(MENU,FUNCTION,PERSONALIZATION,PRINTER,CO  
NCURRENT,LOOKUP,RESPONSIBILITY,FORM,  
DESCFLEX, KEYFLEX, VALUE, PROFILE,  
REQGROUP,  
REQSET,MESSAGE,XML,XDO,IMPORTER): "
```

```
read category
```


Utility Unix – define prompt

```

case $category in
  'REQGROUP')
    parm_prompt="Please enter a request group."
    ;;
  'RESPONSIBILITY')
    parm_prompt="Please enter a responsibility key
(include _ underscore for spaces)."
    ;;
  *)
    parm_prompt="Enter a short name."
esac

```

Utility Unix – user prompt

```
echo $parm_prompt  
read shortname  
if [ -z "$shortname" ] ; then  
    echo $parm_prompt  
    exit  
fi  
respkey=$shortname
```

Utility Unix – user prompt & logfile list

```
echo $parm_prompt
```

```
read shortname
```

```
if [ -z "$shortname" ] ; then
```

```
    echo $parm_prompt
```

```
    exit
```

```
fi
```

```
respkey=$shortname
```

```
## Get initial list of *log files
```

```
ls *log > ${WORK_FILE1} 2>/dev/null
```

Utility Unix – execute FNDLOAD

```
##
```

```
## execute appropriate form of FNDLOAD
```

```
##
```

```
case $category in
```

```
  'MENU')
```

```
    FNDLOAD apps/$PASSWD O Y DOWNLOAD
```

```
    $FND_TOP/patch/115/import/afsload.lct
```

```
    ${shortname}_menu.ldt MENU
```

```
    MENU_NAME=${shortname}
```

```
;;
```

Utility Unix – look for concurrent request logfile

```
## get new list of log files
```

```
ls *log > ${WORK_FILE2}
```

```
## find name of newly created log file (from concurrent  
mgr process)
```

```
FND_LOG=`diff ${WORK_FILE1} ${WORK_FILE2} |  
grep \> | cut -c3-`
```

```
echo "Logfile is:" ${FND_LOG} | tee -a ${LOGFILE}
```

```
## look for obvious success or failure tokens
```

```
grep "successfully" ${FND_LOG} | tee -a ${LOGFILE}
```

```
grep "ORA-" ${FND_LOG} | tee -a ${LOGFILE}
```

Simple Configuration Process

- Following new development, collect all needed objects in staging location
 - .fmb, .rdf, other o/s files
 - Download entities into .ldt files
- Optionally, check-in this staging location (or individual objects) into a version control tool
- Copy or ftp staging location to target site
- Copy o/s level objects into appropriate locations and upload entities into system.

Simple Configuration Process (Con't)

- Use scripts to manage some processes (like download, upload)
- Use spreadsheet, etc, to manage promotion packages and installed locations
- Use other tools as complexity and costs warrant
 - cvs, rccs, sourceforge for version management
 - Quest, HP (Mercury), Applimation, etc

Summary

- Implementing E-Business customizations requires basic objects plus registration steps
- FNDLOAD, java, and adadmin utilities can download and upload needed applications entities
- Use simple scripts to automate portions of configuration management process – add tools as complexity demands