Deployment and Environment Access Guide

Contents

[OVERVIEW 4](#_Toc202644592)

[ENVIRONMENTAL ACCESS AND USAGE 4](#_Toc202644593)

[TEST ENVIRONMENT 4](#_Toc202644594)

[TRAIN ENVIRONMENT 4](#_Toc202644595)

[Access: 4](#_Toc202644596)

[Developer Access: 4](#_Toc202644597)

[PRODUCTION ENVIRONMENT 4](#_Toc202644598)

[Access Policy: 4](#_Toc202644599)

[Deployment Protocol: 4](#_Toc202644600)

[VERIFYING APPLICATION VERSION 5](#_Toc202644601)

[Steps to Check Version: 5](#_Toc202644602)

[Notes: 5](#_Toc202644603)

[PERFORMING A HARD DEPLOYMENT 5](#_Toc202644604)

[ORACLE FORMS DEPLOYMENT & COMPILATION GUIDE 5](#_Toc202644605)

[DIRECTORY STRUCTURE & COMPILATION SETUP 5](#_Toc202644606)

[NAVIGATING the Compilation Directory 5](#_Toc202644607)

[TEST ENVIRONMENTS & SERVER DETAILS 5](#_Toc202644608)

[Common Test Boxes 5](#_Toc202644609)

[SECURE FILE TRANSFER & COMPILATION 6](#_Toc202644610)

[USING SECUREFX 6](#_Toc202644611)

[SECURECRT & SHELL SCRIPT EXECUTION 6](#_Toc202644612)

[LOGGING IN & PREPARING THE ENVIRONMENT 6](#_Toc202644613)

[SHELL SCRIPT DETAILS 6](#_Toc202644614)

[COMPILATION PROCESS 7](#_Toc202644615)

[ADDITIONAL NOTES 7](#_Toc202644616)

[DETERMINING YOUR DISPLAY NAME 7](#_Toc202644617)

[Steps to Find Your Display Name: 7](#_Toc202644618)

[MODIFY DISPLAY VARIABLE 7](#_Toc202644619)

[COMPILATION DIRECTORY SETUP 8](#_Toc202644620)

[Directory Path 8](#_Toc202644621)

[Shell Script 8](#_Toc202644622)

[Do You Need to Build the Form? 8](#_Toc202644623)

[Organizing Files for Compilation 8](#_Toc202644624)

[Compiling Reports with RWConverter 8](#_Toc202644625)

[Permissions and Ownership 9](#_Toc202644626)

[Handling Case Sensitivity in Linux 9](#_Toc202644627)

[Directory Structure Overview 9](#_Toc202644628)

[Notes on Team Collaboration 9](#_Toc202644629)

[Summary 9](#_Toc202644630)

[In Oracle Forms and Reports 10](#_Toc202644631)

[Naming Convention for Parameter Forms 10](#_Toc202644632)

[Where Are These Stored? 10](#_Toc202644633)

[Why It Matters 10](#_Toc202644634)

# OVERVIEW

This guide outlines the access protocols, deployment practices, and environment-specific considerations for development (Dev), testing (Test), and training (Train) environments. It also includes steps for verifying application versions and performing hard deployments.

# ENVIRONMENTAL ACCESS AND USAGE

## TEST ENVIRONMENT

* Purpose: Used for validating changes before production.
* Access: Available to the development team.
* Testing Method:
  + You can log into the test database using the production URL.

## TRAIN ENVIRONMENT

### Access:

* Restricted. No front-end access is granted to developers.
* Reason:
  + Stakeholders and attorneys use this environment to create and demonstrate cases.
  + The data must remain untouched to preserve its integrity.

### Developer Access:

* + Backend (database) access is available but locked.
  + No access to the front-end application, as preferred by stakeholders and respected by the development team.
  + This method was used during the GLS changes deployment.

## PRODUCTION ENVIRONMENT

### Access Policy:

* + Previous leads had limited access (e.g., to reports).
  + The current policy avoids direct developer access to production.

### Deployment Protocol:

* + Always bring down the servers before deploying.
  + Notify users in advance (e.g., “Deployment at 12:00 PM”) to avoid data loss or confusion.
  + Prefer weekend deployments (Saturday/Sunday) to minimize user disruption.

# VERIFYING APPLICATION VERSION

## Steps to Check Version:

1. Log into the application.
2. Click on Help.
3. Select About.
4. Navigate to Attorney Fee section.
5. The version number will be displayed (e.g., "cicd").

### Notes:

The version label (e.g., "cicd") helps identify the deployment stage (e.g., sandbox, Dev).

Update the version label in the form file before deployment to reflect the correct environment.

# PERFORMING A HARD DEPLOYMENT

Tools Used:

* SecureFX: For file transfer and deployment.

Steps:

1. Open the target form in the application.
2. Ensure no changes are made for 35 minutes (heartbeat timeout).
3. Open SecureFX.
4. Navigate to the deployment directory.
5. Copy the updated file (even if already copied, repeat to ensure accuracy).
6. Confirm the file’s Date Modified to verify the latest version.

# ORACLE FORMS DEPLOYMENT & COMPILATION GUIDE

## DIRECTORY STRUCTURE & COMPILATION SETUP

### NAVIGATING the Compilation Directory

* Navigate to the application’s CaseMiss folder.
* Enter the SRC2019 directory:
  + This is the designated folder for compiling Oracle Forms.
  + Ensure this path is included in the environment so forms compile correctly.

## TEST ENVIRONMENTS & SERVER DETAILS

### Common Test Boxes

| **Environment** | **Server Name** | **Notes** |
| --- | --- | --- |
| Sandbox | **vx2** | Also referred to as **WXW**, **BLSS**, **CAST**, **CAS01.ds.test.irs.net.gov** |
| Dev | Listed in sticky notes | Use HTTPS URLs only |
| Test | Shared with Train on same server | Two instances: **test** and **train** |
| Prod | Multiple instances | Includes **Prod1**, **Prod2**, and **Prod Macro** |

* All server details, including URLs and mappings, are documented in the Sticky Notes under “Files Shared with Newcomers.”
* Ignore any references to REL6—these are deprecated.

# SECURE FILE TRANSFER & COMPILATION

## USING SECUREFX

1. Open SecureFX.
2. Navigate to the SRC2019 directory.
3. Copy the compiled form file (e.g., FBN3) into the directory.

* Always use binary mode.
* Overwrite if necessary.

1. Confirm the file’s timestamp (e.g., July 1, 14:26) to ensure the latest version.

# SECURECRT & SHELL SCRIPT EXECUTION

## LOGGING IN & PREPARING THE ENVIRONMENT

* Use SecureCRT 9.x to log in.
* Preferred account: kc\_adm (superuser).
  + Be cautious with shell script edits—minor syntax issues (e.g., extra spaces or carriage returns) can break functionality.
  + Always back up the shell script before making changes.

## SHELL SCRIPT DETAILS

* Script: form\_compile.sh
* Backup the script before editing.
* Modify the DISPLAY variable:
  + Comment out previous user’s display
  + Uncomment and set your own display (e.g., DISPLAY=:0.0).

# COMPILATION PROCESS

* USING REFLECTION X

1. Launch Reflection X to enable GUI-based compilation.
2. Start a new session in SecureCRT.
3. Run the compilation script:

./form\_compile.sh

1. Monitor the output for errors or confirmation of successful compilation.

## ADDITIONAL NOTES

* The kc\_adm account belongs to the oinstall group and has full permission in the compilation directory.
* If the script stops working, consult with the middle-tier team or restore the backup.
* This setup was originally configured under time constraints during the REL6-to-REL8 migration, so improvements may be needed over time.

# DETERMINING YOUR DISPLAY NAME

## Steps to Find Your Display Name:

1. On your workstation, open System Information.
2. Locate the Device Name (e.g., LN001MA445846.ds.test.irs.net.gov).
3. Append :0.0 to the device name to form your display name:

LN001MA445846.ds.test.irs.net.gov:0.0

1. Use this display name in the FRMCMP.sh script for GUI-based compilation.

## MODIFY DISPLAY VARIABLE

1. Open FRMCMP.sh in a text editor.
2. Comment out any existing DISPLAY variable (e.g., Jim’s).
3. Add your own display:

DISPLAY=LN001MA445846.ds.test.irs.net.gov:0.0

1. export DISPLAY

Caution:

* Always back up the script before editing.
* Be careful with whitespace and carriage returns—minor syntax issues can break the script.

# COMPILATION DIRECTORY SETUP

## Directory Path

* Navigate to:
  + /apps/casemus/SRC2019
* This is the designated compilation directory for Oracle Forms.

## Shell Script

* Script name: FRMCMP.sh
* Located in: SRC2019
* This script is used to compile Oracle Forms modules.

## Do You Need to Build the Form?

No, you don’t always need to build the form manually. Oracle Forms modules are compiled individually, and once compiled, they don’t require rebuilding unless changes are made.

## Organizing Files for Compilation

To streamline compilation, group your files into. lstlists:

* **form.lst** – List of all Forms modules
* **lib.lst** – List of library files
* **rwfile.lst** – List of report files for conversion

You can use vi or any text editor to create these lists. Example:

vi form.lst

Then compile all files in the list using a script or loop.

## Compiling Reports with RWConverter

To convert **.rdf** files to **.rep**:

1. Use the **rwconverter** utility.
2. Provide the full path to the .**rdf** file.
3. Example:

rwconverter.sh ... input=full/path/to/file.rdf ...

Even if the ReflectionX server is stopped, the conversion will still work—it’s just a file conversion.

After conversion, verify the .rep file:

ls -lrt \*.rep

## Permissions and Ownership

Ensure correct ownership for compiled files:

* Owner: oracle
* Group: oinstall

If files are owned by other users (e.g., kcadm, ksmus), adjust permissions or ownership as needed.

## Handling Case Sensitivity in Linux

Linux is case-sensitive, unlike Windows. If a menu or form is referenced with different casing, you may need to compile multiple versions:

* Example: COM\_CASE\_MAIN.mmx, com\_case\_main.mmx, Com\_Case\_Main.mmx
* Compile each variation to ensure compatibility with all referencing modules.

## Directory Structure Overview

* Forms Directory: Contains .fmb, .pll, .olb, .mmx, and compiled .fmx files.
* Reports Directory: Contains .rdf and compiled .rep files.
* Libraries:
* .pll – PL/SQL libraries
* .olb – Object libraries
* .mmx – Menu modules

Use ls -lrt to inspect contents:

ls -lrt $APPL\_TOP/forms/US

ls -lrt $APPL\_TOP/reports/US

## Notes on Team Collaboration

* Only two team members had access to the compilation environment.
* Passwords were shared securely and updated promptly.
* New team members unfamiliar with Linux or Oracle Forms faced a learning curve.
* Pairing less experienced users with more tech-savvy colleagues helped bridge the gap.

## Summary

* Use .lst files to batch compile modules.
* Use rwconverter for .rdf to .rep conversion.
* Watch out for case sensitivity on Linux.
* Maintain correct file ownership and permissions.
* Collaborate and document processes for smoother onboarding.

# In Oracle Forms and Reports

In Oracle Forms and Reports, when a report like **GLR\_Caught\_Liability.rdf** has an associated parameter form, it's typically named using a consistent naming convention. Here's how it works:

## Naming Convention for Parameter Forms

If a report requires a parameter form, the parameter form is usually named by appending \_PF (short for "Parameter Form") to the report name.

Example:

* Report file: GLR\_Caught\_Liability.rdf
* Parameter form: GLR\_Caught\_Liability\_PF.fmb

After compilation:

* Compiled report: GLR\_Caught\_Liability.rep
* Compiled parameter form: GLR\_Caught\_Liability\_PF.fmx

## Where Are These Stored?

* .rdf and .rep files → go in the Reports directory
* .fmb and .fmx files → go in the Forms directory

You can verify the presence of parameter forms by checking the forms directory:

ls -lrt $APPL\_TOP/forms/US | grep GLR\_Caught\_Liability\_PF

## Why It Matters

When launching a report that requires user input, Oracle will first call the parameter form (\_PF.fmx) to collect parameters, then pass them to the report (.rep). If the parameter form is missing or misnamed, the report may fail to run properly.