
# KFS Hands-On Developer Introduction

Leo Przybylski

# Contents

# Preface

# Copyright

## Copyright Holder



## Disclaimer



# About the Trainer

Leo started working with the Kuali Foundation in 2005 as a developer on the Kuali Financial System. Since then, he has worked as a *Development Manager* on the Kuali Financial System, *Lead Developer* on the Kuali Coeus project, *Software Architect* on the University of Arizona KFS implementation, and now is a *Release Engineer* for the Kuali Foundation for the Rice Project.

Leo has given six presentations on KFS, KC, and Rice on to separate Kuali Days occassions.

One significant contribution he has made to the Kuali Community is his Rice LDAP Integration module.

# Using these Exercises

## VirtualBox Appliance

Exercise instructions are included in this document. All software and examples are available on the VirtualBox appliance distributed during class. To install the VirtualBox appliance:

1. Execute the VirtualBox installer to install the software.
2. Copy the **KFSDev.box** from the distributed USB drive to your hard disk.

3. Execute from a shell

```
vagrant box add KFSDev KFSDev.box
```

4. Run init

```
vagrant init KFSDev
```

5. Import the Virtual Machine

```
vagrant up
```

6. Stop the Virtual Machine

```
vagrant halt
```

7. Start VirtualBox

## Virtual Machine Manifest

The VirtualBox appliance is an Ubuntu Linux distribution. Within it is the software we will use for this class:

**Eclipse Indigo** the IDE used for class. Includes Subclipse, the m2eclipse plugin, and pre-installed projects with examples.

**OpenJDK 1.7.0_06 IcedTea** the JVM used for executing/testing examples.

**Maven 3** used to build Rice applications, run tests, and start the Tomcat6 application

### Credentials

**User Account** is **kuali** with the password **kuali**. This is used to unlock the VM after it has suspended, gone to sleep, or locked. The password is also required for executing commands as **root** which may on occassion be required. The user account home directory is located at **/home/kuali** and will frequently be referred to during the training.

**Database Account** uses the jdbc connection string **jdbc:mysql://localhost:3306/kuldemo** and the username/password **kuldemo**/**kuldemo**. These are the default credentials and database connection information as defined in kul-cfg-dbs.

### Structure

The Eclipse workspace is located at **/home/kuali/workspace**.

# Training Overview

# Exercise 1:
# Import Project

# Exercise 1
# Import Project

## Description

## Goals

## Instructions

# Notes

# Exercise 2:
# Database Setup

# Exercise 2:
# Database Setup

## Description

## Goals

## Instructions

### 1 Update impex-build.properties

Open **/home/kuali/impex-build.properties**. Locate the section of code that looks like

```
import.torque.database = mysql
import.torque.database.driver = com.mysql.jdbc.Driver
import.torque.database.url = jdbc:mysql://localhost:3306/kuldemo
import.torque.database.user=kuldemo
import.torque.database.schema=KULDEMO
import.torque.database.password=kuldemo
```

and change it to

```
import.torque.database = mysql
import.torque.database.driver = com.mysql.jdbc.Driver
import.torque.database.url = jdbc:mysql://localhost:3306/kuldev
import.torque.database.user=kuldev
import.torque.database.schema=KULdev
import.torque.database.password=kuldev
```

### 1 Update kfs-build.properties

Open **/home/kuali/kfs-build.properties** Locate the section of code that looks like

```
datasource.username=kuldemo
datasource.password=kuldemo
mysql.datasource.url=jdbc:mysql://localhost:3306/kuldemo
```

and change it to

```
datasource.username=kuldev
datasource.password=kuldev
mysql.datasource.url=jdbc:mysql://localhost:3306/kuldev
```

### 2 Run Impex

Using a terminal window do the following:

```
cd /home/kuali/workspace/kul-cfg-dbs/impex
ant create-schema import
```

# Notes

# Exercise 3:
# Create Business Object Table

# Exercise 3:
# Create Business Object Table

## Description

## Goals

## Instructions

### 1 Run mysql

```
mysql -u kuldev -p kuldev
```

### 2 Create Table

```
CREATE TABLE IF NOT EXISTS FERPA_CERTIFICATION_T
  (
     ID          BIGINT(19) NOT NULL,
     PRNCPL_ID   VARCHAR(40) NOT NULL,
     ACTV_IND VARCHAR(1) NOT NULL,
     VER_NBR  DECIMAL(8, 0) DEFAULT 1 NOT NULL,
     OBJ_ID   VARCHAR(36) NOT NULL,
     CONSTRAINT FERPA_CERTIFICATION_T_TC0 UNIQUE (OBJ_ID),
     CONSTRAINT FERPA_CERTIFICATION_T_TP1 PRIMARY KEY(CODE)
  );

CREATE TABLE IF NOT EXISTS FERPA_CERTIFICATION_ID_S
  (
     ID BIGINT(19) NOT NULL AUTO_INCREMENT,
     PRIMARY KEY(ID)
  );
```

# Notes