

Kofax MarkView

Reintegration Guide for Upgrades to Oracle E-Business Suite R12 or 12.2

Version: 10.1.0

Date: 2019-08-21

The KOFAX logo is displayed in a bold, blue, sans-serif font. The letters are thick and closely spaced, with a consistent weight throughout the word.

Portions © 2004-2019 Kofax, 15211 Laguna Canyon Road, Irvine, California 92618, U.S.A. Portions © 2002-2006 Kofax Development GmbH. Portions © 1997-2006 Kofax Development UK Ltd. All Rights Reserved. Use is subject to license terms.

This product is protected by U.S. Patent Nos. 5,159,667 and 7,478,332.

Third-party software is copyrighted and licensed from Kofax's suppliers.

THIS SOFTWARE CONTAINS CONFIDENTIAL INFORMATION AND TRADE SECRETS OF KOFAX USE, DISCLOSURE OR REPRODUCTION IS PROHIBITED WITHOUT THE PRIOR EXPRESS WRITTEN PERMISSION OF KOFAX.

Kofax is a trademark of Kofax, Inc., registered in the U.S. and/or other countries. All other trademarks are the property of their respective owners. No part of this publication may be reproduced, stored, or transmitted in any form without the prior written permission of Kofax.

U.S. Government Rights Commercial software. Government users are subject to the Kofax standard license agreement and applicable provisions of the FAR and its supplements.

You agree that you do not intend to and will not, directly or indirectly, export or transmit the Software or related documentation and technical data to any country to which such export or transmission is restricted by any applicable U.S. regulation or statute, without the prior written consent, if required, of the Bureau of Export Administration of the U.S. Department of Commerce, or such other governmental entity as may have jurisdiction over such export or transmission. You represent and warrant that you are not located in, under the control of, or a national or resident of any such country.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

This product includes software developed by:

- Apache Software Foundation (<http://www.apache.org/>). Licensed under the Apache License, Versions 1.1 and 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-1.1> or <http://www.apache.org/licenses/LICENSE-2.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
- Jean-Loup Gailly and Mark Adler - Copyright © 1995-2004.
- The JDOM Project (<http://www.jdom.org/>).
- Jaxen Project (<http://jaxen.org/>). Copyright © 2001 werken digital. All Rights Reserved.

Copyright © 2000 – 2006 The Legion Of The Bouncy Castle (<http://www.bouncycastle.org>)

All other third-party products or brands mentioned in this documentation are or may be trademarks or registered trademarks and are the property of their respective owners.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

ARE DISCLAIMED. IN NO EVENT SHALL BIT MIRACLE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Table of Contents

Preface.....	6
Who should read this guide.....	6
Related documentation.....	6
Getting help with Kofax products.....	7
Chapter 1: Reintegrate MarkView with Oracle.....	9
Enable editions (for Oracle EBS 12.2 only).....	9
Assign grants and privileges to the MarkView schema user.....	9
Recreate grants and synonyms.....	10
Compile invalid objects in the MarkView schema.....	10
Re-enable triggers in the MarkView schema.....	10
Migrating MarkView data to Oracle EBS 12.x format.....	11
Reconfigure Self-Service Invoice (SSI).....	11
Recreate Oracle EBS customizations (Oracle EBS 12.2 only).....	11
Chapter 2: Reinstall MarkView Oracle Objects.....	13
Redefine c_MARKVIEW_TOP (Oracle EBS 12.2 only).....	13
Redefine c_MARKVIEW_TOP (Oracle EBS 12.1 and earlier).....	14
Set up c_MARKVIEW_TOP in Oracle E-Business Suite.....	15
Run the MarkView Oracle Objects installer.....	16
Add the log4j logging library to the class path (Oracle EBS 12.2 only).....	17
Chapter 3: Configure Oracle Forms.....	18
Verify the environment (Oracle EBS 12.2 only).....	18
Test the \$c_MARKVIEW_TOP environment variable.....	18
Set up the Oracle Forms environment variables.....	19
Start Oracle Forms Builder.....	20
Update the Oracle Attachments form (FNDATTCH.fmb).....	20
Update the custom library (CUSTOM.pll).....	24
Enable Invoice Workbench (optional).....	27
Configure Oracle Forms for multi-language support (optional).....	31
Enable MarkView Quick Invoices (optional).....	31
Remove MarkView personalizations from Supplier form (optional).....	36
Synchronize the run and patch file systems (Oracle EBS 12.2 only).....	36
Synchronize the MarkView files using adop phase=fs_clone command.....	36
Synchronize the MarkView files using custom synchronization driver (for UNIX only).....	37
Chapter 4: Expense Management Configuration.....	38

Configure Oracle Expenses workflow.....	38
Set the administrator access level.....	38
Open the Expenses workflow.....	39
Add functions to the workflow process.....	40
Configure the workflow background process.....	48
Configure the missing-receipt timeout.....	50
Chapter 5: Oracle configuration parameters.....	53
Enable Tax Classification Codes.....	53
Add the Automatic Distributions check box.....	53
Set default values for the Automatic Distributions check box.....	54
Set default values for Tax fields.....	55
Set maximum values for frequently used accounts.....	55
Enable descriptive flex fields.....	56
Enable project fields.....	56
Enable Income Tax.....	57
Add Undistributed Amount.....	58
Use tax default hierarchy.....	58
Allow post validation changes.....	59
Enable posted distribution lines changes.....	59
Use withholding tax default hierarchy.....	60
Custom account validation.....	61
Chapter 6: Make other updates after applying an Oracle Applications patch.....	62
Redo modifications.....	62
Reapply modifications to SFXINWKB.....	62
Reapply modifications to Expense.....	62
Observe best practices.....	63
Appendix A: Third-party license agreement.....	64

Preface

This guide describes how to reintegrate MarkView after you upgrade Oracle E-Business Suite to R12 or 12.2. This process includes:

- Reintegrating MarkView with Oracle; recreating grants, synonyms, and triggers; migrating data; and reconfiguring Self-Service Invoice. See [Reintegrate MarkView with Oracle](#).
- [Reinstall MarkView Oracle Objects](#)
- [Configure Oracle Forms](#)
- [Expense Management Configuration](#)

Note Only MarkView 8.2 and higher supports Oracle E-Business Suite 12.2. If you have an environment with an earlier version of MarkView installed, upgrade to MarkView 8.2 or higher first. See the instructions in the *MarkView Upgrade Guide*.

Who should read this guide

Use the information in this document if your site upgraded the Oracle E-Business Suite version to R12 or higher. Reintegrating the Oracle E-Business Suite with MarkView requires the following areas of expertise:

- Oracle Database Administrators who are experienced with Oracle Relational Database Management System (RDBMS), can edit and run PL/SQL scripts, and are familiar with their ERP system.
- Application Server Administrators who have J2EE technology knowledge.
- Windows Server Administrators who have Microsoft Windows experience and who can install and configure Windows applications and hardware.
- Customers who are planning a new installation or a migration from a previous release of any MarkView product.

Administrators should also be familiar with MarkView system operation.

Related documentation

The documentation set for Kofax MarkView is available online:¹

<http://docshield.kofax.com/Portal/Products/MarkView/10.1.0-zkxlep5jk7/MarkView.htm>

In addition to this guide, the documentation set includes the following items:

¹ You must be connected to the Internet to access the full documentation set online. If the security policy for your organization requires offline access (without an Internet connection), see the Installation Guide.

Kofax MarkView Features Guide

Use this guide to learn about the features included and options available with MarkView; to become familiar with MarkView products; and to decide which are important to the business challenges you face and best suit your site. This guide includes information about how features impact the workflow, the interaction between features, the touch points with the ERP system, and how features address business problems.

Kofax MarkView Planning Guide

Use this guide to learn about the prerequisites for implementing MarkView products. This guide includes system information, such as the protocols required for communication between servers, hardware and software prerequisites, and minimum RAM requirements.

Use this guide in conjunction with the *Technical Specifications* document on the [Kofax website](#) to prepare a site for product installation.

Kofax MarkView Installation Worksheet

Use this worksheet to collect and record the information you need to install or upgrade MarkView products.

Kofax MarkView Installation Guide

Use this guide in conjunction with the *Kofax MarkView Installation Worksheet* to install and configure MarkView products and to configure third-party products that integrate with MarkView.

Kofax MarkView Upgrade Guide

Use this guide in conjunction with the *Kofax MarkView Installation Worksheet* to upgrade and configure MarkView products.

Kofax MarkView Administrator's Guide, Volume 1

Use this guide to administer the MarkView system. This guide describes how to configure and maintain the applications, solutions, and users that make up the MarkView Suite. The guide also describes how MarkView influences the administration of other servers and software that interface with MarkView applications.

The MarkView Administrator should be well-versed in database administration, application server setup, tuning and maintenance, or should know where to get such information. The administrator's guide does not replicate this information, but conveys MarkView product-specific information.

Kofax MarkView Administrator's Guide, Volume 2

Use this guide to maintain MarkView components that are administered outside of the MarkView interface. This guide includes advanced administrative tasks and describes MarkView custom packages and join points.

Kofax MarkView Release Notes

Use this document to learn what is new with the latest MarkView release, identify outstanding defects and workaround solutions where applicable, and learn which defects the release fixes.

Getting help with Kofax products

The [Kofax Knowledge Base](#) repository contains articles that are updated on a regular basis to keep you informed about Kofax products. We encourage you to use the Knowledge Base to obtain answers to your product questions.

To access the Kofax Knowledge Base, go to the Kofax [website](#) and select **Support** on the home page.

Note The Kofax Knowledge Base is optimized for use with Google Chrome, Mozilla Firefox or Microsoft Edge.

The Kofax Knowledge Base provides:

- Powerful search capabilities to help you quickly locate the information you need.
Type your search terms or phrase into the **Search** box, and then click the search icon.
- Product information, configuration details and documentation, including release news.
Scroll through the Kofax Knowledge Base home page to locate a product family. Then click a product family name to view a list of related articles. Please note that some product families require a valid Kofax Portal login to view related articles.
- Access to the Kofax Customer Portal (for eligible customers).
Click the **Customer Support** link at the top of the page, and then click **Log in to the Customer Portal**.
- Access to the Kofax Partner Portal (for eligible partners).
Click the **Partner Support** link at the top of the page, and then click **Log in to the Partner Portal**.
- Access to Kofax support commitments, lifecycle policies, electronic fulfillment details, and self-service tools.
Scroll to the **General Support** section, click **Support Details**, and then select the appropriate tab.

Chapter 1

Reintegrate MarkView with Oracle

Extract the zip file for your version of MarkView. You can place the files on any machine with access to the database server. Using the scripts provided in the zip file, complete the tasks in this chapter.

1. Log in to the [Kofax Fulfillment Site](#).
2. From the Your Software list, select **markview-x.x.x_oracle.zip** where x.x.x is the MarkView version number.
3. Extract the zip file onto any system with access to the database server.

Enable editions (for Oracle EBS 12.2 only)

Note Only enable editions if you are integrating MarkView with Oracle E-Business Suite 12.2 or higher. Running this script with earlier versions of Oracle E-Business Suite will corrupt MarkView.

Use `enable_editions.sql` to make the MarkView schema editionable. Do not modify this script.

1. Verify that MarkView 8.2 or higher has been applied and that the upgrade to Oracle E-Business Suite 12.2 is completed.
2. Log in to the MarkView database as SYSDBA.
3. Navigate to `<distribution_directory>\installer\scripts`, where `<distribution_directory>` is the location where you downloaded and decompressed the installation files.
4. Run `enable_editions.sql`.
5. At the system prompt for the user name, enter the MarkView database schema name, for example, `markview`.

Assign grants and privileges to the MarkView schema user

The `markview_grant_privileges.sql` script assigns grants and privileges to the schema user. Do not modify this script.

1. Log in to the MarkView database as SYSDBA.
2. Navigate to `<distribution_directory>\installer\scripts`, where `<distribution_directory>` is the location where you downloaded and decompressed the installation files.
3. Run `markview_grant_privileges.sql`.
4. At the prompt, enter the MarkView schema user name.

Recreate grants and synonyms

The following steps apply only after upgrading to Oracle E-Business Suite 12.2.

1. Log in to SQL*Plus and set the working directory to **modules/erp-integration-oracle-dist-x.x.x/oracle-toolkit/sql/schema/grants_synonyms** where x.x.x is the module version number.
For MarkView 9.1, use this directory: `modules/erp-integration-oracle-dist-9.1.0/oracle-toolkit/sql/schema/grants_synonyms`
2. At the SQL prompt, run the script **@mvoa_create_grants_synonyms.sql**.
3. Follow the prompts to enter the MarkView and Apps schema information.
4. When prompted for the Oracle E-Business Suite version, enter **11.5** (select this version for Oracle EBS R12 and 12.2).

Note When granting access on FND_DOCUMENTS, FND_DOCUMENTS_TL, AP_TAX_DEFAULT_PKG, AP_TAX_RECOVERY_PKG to MarkView, error messages may appear. This is expected behavior. The error messages can be ignored.

Compile invalid objects in the MarkView schema

1. Log in to SQL*Plus as the MarkView schema owner and set the working directory to **scripts**.
2. At the SQL prompt, run **@gencomp.sql**.
You see a message similar to the following:

```
Recursively compiling 2 object(s).
Please wait...
All objects compiled successfully.
```

3. Review the information in the COMPILER.LOG. If the message says that it is recursively compiling more than 0 objects, rerun **@gencomp.sql**. Dependencies between objects can require one to complete compilations successfully before the other completes compilations.

Re-enable triggers in the MarkView schema

Re-enable all MarkView triggers.

1. Log in to SQL*Plus as the MarkView schema owner.
2. At the SQL prompt, run these commands:

```
col fullline form a80
set pages 0
set feed off
spool enable.sql
select 'set echo on' from dual;
select 'spool enable.log' from dual;
select 'alter trigger ' || trigger_name || ' enable;'
from user_triggers
where status = 'DISABLED';
select 'spool off' from dual;
select 'set echo off' from dual;
```

```
spool off
set pages 66
set feed on
@enable
```

Migrating MarkView data to Oracle EBS 12.x format

1. Log in to SQL*Plus as the Oracle E-Business Suite schema owner (APPS) and set the working directory to **modules\erp-integration-oracle-dist-x.x.x\erp-integration-oracle-db\patches\apps** where x.x.x is the module version number.
For MarkView 9.1, use this directory: modules/erp-integration-oracle-dist-9.1.0/erp-integration-oracle-db/patches/apps
2. At the SQL prompt, run **@oracle-integ-apps-1000-update-datatype-category.sql**.
3. At the SQL prompt, run **@oracle-integ-apps-1001-update-file-name.sql**.
4. At the SQL prompt, run **@oracle-integ-apps-1002-update-negative-filenames.sql**.

Reconfigure Self-Service Invoice (SSI)

If you are changing to Regime to Rate tax entry, open MarkView Administration and change the preference value for SSI_TAX_ENTRY_MODE to REGIME_TO_RATE.

If you use SSI Templates that contain tax information, use the SSI interface to re-create the templates. Create new invoice requests that contain all of the information to be in the template including tax. Save the new request and overwrite the existing templates.

Recreate Oracle EBS customizations (Oracle EBS 12.2 only)

Upgrade to Oracle EBS 12.2 may clear some customizations created by the MarkView installer. Use the scripts below to recreate these customizations.

1. Log in to the database as the APPS user.
2. Update Oracle EBS to use the customized Invoice Workbench form by running the following script:

```
declare
  n_orig_form_id      Number;
  n_sf_form_id        Number;
  n_application_id    Number;
begin
  select form_id
  into n_orig_form_id
  from fnd_form
  where form_name = 'APXINWKB';

  select form_id, application_id
  into n_sf_form_id, n_application_id
  from fnd_form
  where form_name = 'SFXINWKB';

  update fnd_form_functions
  set form_id = n_sf_form_id,
```

```
        application_id = n_application_id
      where form_id = n_orig_form_id;
end;
/
```

3. Update Oracle EBS to work with the MarkView attachments by running the following script:

```
update fnd_document_entities
  set pk1_column = 'INVOICE_ID'
  where data_object_code = 'AP_INVOICES_INTERFACE'
     and table_name = 'AP_INVOICES_INTERFACE';
commit;
```

Chapter 2

Reinstall MarkView Oracle Objects

Follow these steps to reinstall the MarkView Oracle Objects:

1. Redefine the `c_MARKVIEW_TOP` environment variable. See [Redefine c_MARKVIEW_TOP \(Oracle EBS 12.1 and earlier\)](#)
2. Set up `c_MARKVIEW_TOP` in Oracle. See [Set up c_MARKVIEW_TOP in Oracle E-Business Suite](#)
3. Run the Oracle Objects installer. See [Run the MarkView Oracle Objects installer](#)

Redefine `c_MARKVIEW_TOP` (Oracle EBS 12.2 only)

Note Only follow the steps below if you upgraded to Oracle E-Business Suite 12.2.

Skip this section if you upgrade from MarkView 8.2 and higher integrated with Oracle E-Business Suite 12.2 and continue at [Set up c_MARKVIEW_TOP in Oracle E-Business Suite](#).

Redefine the `c_MARKVIEW_TOP` environment variable:

1. Remove all the occurrences of `c_MARKVIEW_TOP` from `FND_OAM_CONTEXT_CUSTOM` table:
 - a. Log in to the database as the APPS user.
 - b. From the SQL prompt, run the following script:

```
delete from fnd_oam_context_custom where  
name = 'c_MARKVIEW_TOP' or name = 'C_MARKVIEW_TOP'; commit;
```

2. Remove all the occurrences of `c_MARKVIEW_TOP` from `$CONTEXT_FILE` XML file:

- a. On the Oracle E-Business Suite server, open a command shell and log in as the Oracle E-Business Suite owner. The default account for this user is `applmgr`.
- b. To set the environment variables, source the run file system environment file.
For example, from a UNIX shell, run the following command:

```
$ source $APPL_TOP/../../../../EBSapps.env run
```

- c. Open `$CONTEXT_FILE` XML file and remove all the lines which contain `'c_MARKVIEW_TOP'`.
For example, remove the following line, if any:

```
<c_MARKVIEW_TOP oa_var="c_MARKVIEW_TOP" scope="CUSTOM" oa_type="PROD_TOP"
```

```
oa_enabled="FALSE">/appl01/apps/apps_st/appl/markview/12.0.0</c_MARKVIEW_TOP>
```

- d. To set the environment variables, source the patch file system environment file.

For example, from a UNIX shell, run the following command:

```
$ source $APPL_TOP/../../../../EBSapps.env patch
```

- e. Open \$CONTEXT_FILE XML file and remove all the lines which contain 'c_MARKVIEW_TOP'.
- f. To set the environment variables, source the run file system environment file.

For example, from a UNIX shell, run the following command:

```
$ source $APPL_TOP/../../../../EBSapps.env run
```

3. Add the c_MARKVIEW_TOP environment variable to the \$customfile custom environment files.

- a. On the Oracle E-Business Suite server, open a command shell and log in as the Oracle E-Business Suite owner. The default account for this user is applmgr.

- b. To set the environment variables, source the run file system environment file.

For example, from a UNIX shell, run the following command:

```
$ source $APPL_TOP/../../../../EBSapps.env run
```

- c. Create the \$customfile custom environment file (if it does not already exist).

- d. Add the following lines to the end of the \$customfile custom environment file:

```
export c_MARKVIEW_TOP=`dirname ${BASH_SOURCE[1]}`/markview/12.0.0
export C_MARKVIEW_TOP=$c_MARKVIEW_TOP
```

- e. To set the environment variables, source the patch file system environment file.

For example, from a UNIX shell, run the following command:

```
$ source $APPL_TOP/../../../../EBSapps.env patch
```

- f. Create the \$customfile custom environment file (if it does not already exist).

- g. Add the following lines to the end of the \$customfile custom environment file:

```
export c_MARKVIEW_TOP=`dirname ${BASH_SOURCE[1]}`/markview/12.0.0
export C_MARKVIEW_TOP=$c_MARKVIEW_TOP
```

- h. To set the environment variables, source the run file system environment file.

For example, from a UNIX shell, run the following command:

```
$ source $APPL_TOP/../../../../EBSapps.env run
```

Redefine c_MARKVIEW_TOP (Oracle EBS 12.1 and earlier)

Note Do not follow the steps below if you upgraded to Oracle E-Business Suite 12.2.

Redefine the c_MARKVIEW_TOP environment variable:

1. Log in to Oracle E-Business Suite as System Administrator.
2. Navigate to **System Administrator > Oracle Application Manager > OAM Setup > Site Map > AutoConfig**.
3. The Applications Dashboard opens the **Context Files** window. Find the Related Links section at the bottom, and click **Manage Custom Parameters**.

- On the next page (Customized Parameters), click **Add**.

Note If you see the error message: "Customization is not allowed on different versions of configuration files.", see Oracle MetaLink 762590.1: Create a Custom Parameter Fails With - Customization Is Not Allowed On Different Versions of Configuration Files.

- Select **Applications Tier** and click **Next**.
- Set up the next page as follows:

Field	Description and Setting
OA_VAR	The name of the environment variable. Select c_MARKVIEW_TOP.
Default Value	The c_MARKVIEW_TOP directory. For installation set to: %s_at%/markview/12.0.0 (%s_at% is APPL_TOP). For upgrade set to %s_at%/<your TOP name>/12.0.0 (Use 12.0.0 for R12.1)
Title	MarkView Top
Description	MarkView for Oracle Forms Top
OA Type	PROD_TOP

- On the Step 2 page, click **Next**.
- On the confirmation page verify your values and click **Next**.
Step 4 summarizes the files that change when you change the TOP variable. You can read the XML file listed here in your shell by viewing \$CONTEXT_FILE.
- Click **Finish**.
- Go to [Set up c_MARKVIEW_TOP in Oracle E-Business Suite](#) on page 15.
AutoConfig loads the contents of \$CONTEXT_FILE into your shell and the Oracle Forms environment via \$ORACLE_CONFIG_HOME/forms/server/default.env.

Set up c_MARKVIEW_TOP in Oracle E-Business Suite

- Run AutoConfig to load c_MARKVIEW_TOP into your environment and ensure that future runs of AutoConfig will not unset c_MARKVIEW_TOP. See Oracle documentation for more detailed information on AutoConfig.
On the Oracle E-Business Suite server, open a command shell and log in as the Oracle E-Business suite owner. The default account for this user is applmgr. Run the AutoConfig script:
 - `$INST_TOP/admin/scripts/adautocfg.sh`
- After AutoConfig finishes running, stop and start Oracle Applications.
- Confirm that the variable is set:
 - Log out of your shell and close your remote connection.
 - Log in to the shell and run `echo $c_MARKVIEW_TOP` to display the variable.

Note If after running AutoConfig and logging into your shell, the variable is not available, see MetaLink Note: 461326.1 for a bug that may apply to your system.

Run the MarkView Oracle Objects installer

To run the installer:

1. On the Oracle E-Business Suite server, open a command shell and log in as the Oracle E-Business suite owner. The default account for this user is applmgr.
2. Make sure that all environment variables are correctly set up for this account, check:
 - \$PATH: Confirm the appropriate Java version comes first (or run `java -version`). For the Java version, see the *Technical Specifications* document on the [Kofax website](#).
 - \$APPL_TOP: Confirm this is the correct instance of Oracle E-Business Suite
 - \$FORMS_PATH: Confirm that it includes \$AU_TOP/resource and \$AU_TOP/forms/US.
3. Ensure that the application server is running.
4. Navigate to `<distribution_directory>/installer/bin`, where `<distribution_directory>` is the temporary distribution directory.
5. Start the installer by running either:
 - For Unix: `./install.sh`
 - For Windows: `install.bat`
6. Select the applications to install. Available choices are:
 - Oracle Forms Integration
 - OA Framework Integration for AP
 - OA Framework Integration for Expense

Consider the following when selecting products:

- Only systems running Oracle Release 12 that also include MarkView for Account Payables can select OA Framework Integration for AP.
 - Only systems running MarkView for Expense Management can select OA Framework Integration for Expense.
7. Click **Next**.
 8. Select **Standard Full Install** and click **Next**.
 9. Click **Select Folder** to navigate to the MarkView installation directory and click **Next**.
 10. Enter your MarkView schema password and click **Next**.
 11. Enter your Oracle applications schema password and click **Next**.
The order and appearance of the next few steps vary depending on the installation options.
 12. Review the values displayed and click **Next**:
 - **Location of APPL_TOP** value: Confirm this is the instance of Oracle E-Business Suite where you want to install Forms.
 - **Location of AU_TOP and FND_TOP** values: Confirm this is the instance of Oracle E-Business Suite where you want to install Forms (both directories should be under \$APPL_TOP).
 - **Location of \$c_MARKVIEW_TOP** value: It should be the value you set up in the Pre-Install steps.

13. Verify the Oracle Framework Environment Variable values APPL_TOP, COMMON_TOP, and JAVA_TOP and click **Next**.
14. Review the Summary and click **Next**.
15. Click **Install**.
The process can take several minutes. To track the progress, click **Show Details**.
The installation finishes with a message indicating whether the installation is successful.
The installation script creates log files in the MarkView installation directory.
16. Click **Exit**.

Add the log4j logging library to the class path (Oracle EBS 12.2 only)

After running the Oracle Objects installer on Oracle EBS 12.2, manually add the log4j library to the class path. If you do not add the log4j library, you may encounter errors while working with Oracle Forms, for example an unexpected error may occur in the iExpenses module.

1. Extract the \$JAVA_TOP/log4j.jar file into a temporary directory.

For example, from a UNIX shell, run the following commands:

```
$ cd $JAVA_TOP
$ mkdir /tmp/log4j
$ cp -p log4j.jar /tmp/log4j
$ cd /tmp/log4j
$ jar xf log4j.jar
```

2. Move or copy the org folder from the temporary directory to \$JAVA_TOP. The unpacked org directory contains one folder called log4j.

For example, from a UNIX shell, run the following command:

```
$ mv /tmp/log4j/org $JAVA_TOP
```

3. Run the Oracle Adadmin utility to generate the new jar files.
 - a. From the Oracle Adadmin utility, select option **1: Generate Applications Files menu**.
 - b. Select option **4: Generate product JAR files**.
 - c. If you wish to force regeneration of all jar files, when prompted, enter **Y**.

Continue at [Configure Oracle Forms](#) on page 18 to configure the MarkView Oracle Forms Server for use with MarkView products.

Chapter 3

Configure Oracle Forms

If you have made customizations to the CUSTOM.pll file, follow the steps in this chapter:

- [Verify the environment \(Oracle EBS 12.2 only\)](#)
- [Set up the Oracle Forms environment variables](#)
- [Start Oracle Forms Builder](#)
- [Update the custom library \(CUSTOM.pll\)](#)
- [Synchronize the run and patch file systems \(Oracle EBS 12.2 only\)](#)

Verify the environment (Oracle EBS 12.2 only)

Follow these steps if you are integrating MarkView with Oracle E-Business Suite 12.2 or higher

When configuring Oracle Forms, run any shell commands on the run file system.

1. On the Oracle E-Business Suite server, open a command shell and log in as the Oracle E-Business Suite owner.

The default account for this user is applmgr.

2. Source the run file system environment file.

For example, from a UNIX shell, run the following command:

```
$ source $APPL_TOP/../.././EBSapps.env run
```

3. Verify that the environment is properly set by examining the relevant environment variable.

For example, from a UNIX shell, run the following command:

```
$ echo $FILE_EDITION
run
```

Test the \$c_MARKVIEW_TOP environment variable

1. Log in to Oracle Applications and open any form (except Invoice Workbench).
2. Select **Help > Diagnostics > Examine**.
3. When prompted, enter the <password>.
4. Navigate to **Block** and press **CTRL-L**.
5. Select **\$ENVIRONMENT\$**.

6. In **Field**, enter one of the following commands and click **Tab**:

- For servlet mode, enter: **C_MARKVIEW_TOP**
- For socket mode, enter: **c_MARKVIEW_TOP**

Note The command is case-specific.

Note Do not press CTRL-L to display the list of values.

7. Verify the **Value** setting.

- If the pathname for either the c_MARKVIEW_TOP variable or the C_MARKVIEW_TOP variable appears in the field, the variable is configured correctly.
- If the pathname does not appear for either c_MARKVIEW_TOP or C_MARKVIEW_TOP, try steps 1-6 again, but use the alternative case. For example, if you initially used c_MARKVIEW_TOP, try C_MARKVIEW_TOP.
- If the pathname does not appear for either c_MARKVIEW_TOP or C_MARKVIEW_TOP after using the alternative case, there is a configuration error. Check if the variable was configured correctly.

8. Verify the value in the database:

- a. Log in to the database as the APPS user.
- b. Run the following SQL command:

```
select basepath from fnd_application where
       basepath like '%MARKVIEW_TOP%';
```

- c. If the return value corresponds to the variable that you verified in step 7, continue at [Set up the Oracle Forms environment variables](#).
- d. If the return value does not correspond to the variable that you verified in step 7, there may be an error when opening a MarkView form such as Invoice Workbench (SFXINWKB).
- e. From the SQL prompt, update the base path to correspond to the environment. Enter the correct code for the value that you verified in step 7:

- For c_MARKVIEW_TOP, enter:

```
update fnd_application set basepath= 'c_MARKVIEW_TOP' where
       basepath like '%MARKVIEW_TOP%'; commit;
```

- For C_MARKVIEW_TOP, enter:

```
update fnd_application set basepath= 'C_MARKVIEW_TOP' where
       basepath like '%MARKVIEW_TOP%'; commit;
```

Set up the Oracle Forms environment variables

Before launching Oracle Forms Builder, set environment variables. Log in to the Forms host as a user with write privileges in the \$APPL_TOP directory (for example, applmgr).

Oracle Forms Builder must run through a windowing system. Initiate this session on a desktop environment or from an X Windows client.

For X Windows clients, set the DISPLAY environment variable to the IP address of your local machine. For example:

```
export DISPLAY=xxx.xxx.xxx.xxx:0
```

The \$APPL_TOP directory contains configuration files which have the suffix .env and set environment variables. These environment variables must be set properly so you can compile forms and libraries.

Before invoking Oracle Forms Builder, you must invoke the appropriate configuration file to establish the proper working environment. Use the shell command that runs scripts or batch files in your current environment:

- C shell: source APPS<sid>_<hostname>.env
- KornShell: APPS<sid>_<hostname>.env

where <sid> is the database SID and <hostname> is the server name.

The configuration file establishes several environment variables, including the following:

Variable	What it Holds
APPL_TOP	The name of the directory where the configuration files reside.
FORMS_PATH in R12.x	One or more directories required to compile forms and libraries.
AU_TOP	The source code for all forms.

To verify FORMS_PATH:

1. Check FORMS_PATH to make sure that it includes \$AU_TOP/resource.
2. Oracle Forms Builder must have \$AU_TOP/forms/US in its path. Either:
 - Launch Oracle Forms Builder from the directory \$AU_TOP/forms/US.
 - Set the variable FORMS_PATH to include \$AU_TOP/forms/US.

Start Oracle Forms Builder

Before running the Oracle Forms Builder installer, verify that the Java version on the system matches the Oracle EBS Java (32-bit or 64-bit versions).

To start Oracle Forms Builder for Oracle Release 12, enter the following command in a shell:

```
frmbld
```

Update the Oracle Attachments form (FNDATTCH.fmb)

By default, Oracle Applications support attachments on most standard Oracle Applications forms. A user clicks the Attachments (paper clip) button on the Oracle Applications toolbar, which starts the FNDATTCH form. The FNDATTCH form searches for and returns any attachments to the current Oracle Applications record. Modify the FNDATTCH form so that it shows attachments with MarkView categories.

Many of the changes to the FNDATTCH form are encapsulated in a source form named 170ATTCH that is installed by the MarkView installer. As you update the FNDATTCH form, cut and paste some of the MarkView modifications from 170ATTCH to FNDATTCH.

Step 1: Open the FNDATTCH and 170ATTCH forms

1. Copy:

`$AU_TOP/forms/US/FNDATTCH.fmb`

to:

`$_c_MARKVIEW_TOP/forms/US`

2. Start Oracle Forms Builder as described in [Start Oracle Forms Builder](#) on page 20.
3. Connect to the Oracle APPS database schema.
 - a. Select **File > Connect**.
 - b. Enter the Oracle APPS database user, password, and connect string (the same credentials specified during MarkView for Oracle Applications installation).
4. Select **File > Open** and open `$_c_MARKVIEW_TOP/MVOA/<version_number>/forms/170ATTCH.fmb` (where `<version_number>` is the Oracle version number). The name of the form, ATTCH170, appears at the top of the navigation tree.
5. Select **File > Open** and open `$_c_MARKVIEW_TOP/forms/US/FNDATTCH.fmb`. The name of the form, FNDATTCH, appears at the top of the navigation tree.

Step 2: Paste the GET_MV_DOCUMENT_ID item into the FNDATTCH form

1. In the **Object Navigator**, expand **ATTCH170 > Data Blocks > DOCUMENT_CONTROL > Items** and select **GET_MV_DOCUMENT_ID**.
2. Right-click and select **Copy**.
3. In the **Object Navigator**, expand **FNDATTCH > Data Blocks > DOCUMENT_CONTROL** and select **Items**.
4. Right-click and select **Paste**.

Step 3: Paste the MarkView package into the FNDATTCH form

1. In the **Object Navigator**, expand **ATTCH170 > Program Units** and highlight these packages:
 - **MARKVIEW* (Package Spec)**
 - **MARKVIEW* (Package Body)**
2. Right-click and select **Copy**.
3. In the **Object Navigator**, expand **FNDATTCH** and select **Program Units**.
4. Right-click and select **Paste**.

Step 4: Edit the PRE-FORM trigger

1. In the **Object Navigator**, expand **FNDATTCH > Triggers** and select **PRE-FORM**.
2. Right-click and select **PL/SQL Editor**.

3. Make the following changes to the PRE-FORM trigger:

- a. Append **.170** to the revision number to indicate that the form is edited for MarkView integration.
For example, change:

```
FND_STANDARD.FORM_INFO('$Revision: 115.155.11591.2')
```

to:

```
FND_STANDARD.FORM_INFO('$Revision: 115.155.11591.2.170')
```

- b. Add the following code to the end of the trigger as a separate line:

```
MarkView.Event('PRE-FORM');
```

4. **Program > Compile PL/SQL > Incremental** and close the window.

Step 5: Edit the PRE-QUERY trigger

1. In the **Object Navigator**, expand **FNDATTCH > Data Blocks > DOCUMENT_HEADER > Triggers** and select **PRE-QUERY**.
2. Right-click and select **PL/SQL Editor**.
3. In the PL/SQL Editor, add the following line of code to the end of the trigger:

```
MarkView.Initialize;
```

4. **Program > Compile PL/SQL > Incremental** and close the window.

Step 6: Edit the POST-QUERY trigger

1. In the **Object Navigator**, expand **FNDATTCH > Data Blocks > DOCUMENT_HEADER > Triggers** and select **POST-QUERY**.
2. Right-click and select **PL/SQL Editor**.
3. Add the following line of code to the end of the trigger:

```
MarkView.Event('POST-QUERY');
```

4. **Program > Compile PL/SQL > Incremental** and close the window.

Step 7: Edit the PRE-INSERT trigger

1. In the **Object Navigator**, expand **FNDATTCH > Data Blocks > DOCUMENT_HEADER > Triggers** and select **PRE-INSERT**.
2. Right-click and select **PL/SQL Editor**.
3. Add the following code to the beginning of the trigger:

```
If MarkView.IsMarkView then  
MarkView.Event('PRE-INSERT');  
End if;
```

4. **Program > Compile PL/SQL > Incremental** and close the window.

Step 8: Edit the WHEN-NEW-RECORD-INSTANCE trigger

1. In the **Object Navigator**, expand **FNDATTCH > Data Blocks > DOCUMENT_HEADER > Triggers** and select **WHEN-NEW-RECORD-INSTANCE**.
2. Right-click and select **PL/SQL Editor**.

3. In the **PL/SQL Editor**, add the following code to the end of the trigger:

```
MarkView.Default_Show_Document;
```

4. **Program > Compile PL/SQL > Incremental** and close the window.

Step 9: Create the KEY-NEXT-ITEM trigger

1. In the **Object Navigator**, expand **FNDATTCH > Data Blocks > DOCUMENT_HEADER > Items > DATATYPE_NAME** and select **Triggers**.
2. Click the plus sign **+** button in the toolbar to create a new trigger.
A list of trigger names appears.
3. From the list, select **KEY-NEXT-ITEM** and click **OK**.
The PL/SQL Editor opens.
4. Add the following code to the new KEY-NEXT-ITEM trigger:

```
MarkView.Event('KEY-NEXT-ITEM');
```

5. **Program > Compile PL/SQL > Incremental** and close the window.

Step 10: Edit the WHEN-VALIDATE-ITEM trigger for the DATATYPE_NAME item

1. In the **Object Navigator**, expand **FNDATTCH > Data Blocks > DOCUMENT_HEADER > Items > DATATYPE_NAME > Triggers** and select **WHEN-VALIDATE-ITEM**.
2. Right-click and select **PL/SQL Editor**.
3. Append the following code to the end of the trigger:

```
MarkView.Event('WHEN-VALIDATE-ITEM');
```

4. **Program > Compile PL/SQL > Incremental** and close the window.

Step 11: Edit the WHEN-VALIDATE-ITEM trigger for the CATEGORY_DESCRIPTION item

1. In the **Object Navigator**, expand **FNDATTCH > Data Blocks > DOCUMENT_HEADER > Items > CATEGORY_DESCRIPTION > Triggers** and select **WHEN-VALIDATE-ITEM**.
2. Right-click and select **PL/SQL Editor**.
3. Append the following code to the end of the trigger:

```
MarkView.Event('WHEN-VALIDATE-ITEM');
```

4. **Program > Compile PL/SQL > Incremental** and close the window.

Step 12: Add an alert to the FNDATTCH form

1. In the **Object Navigator**, expand **ATTCH170 > Alerts** and select **MVOA_MESSAGE**.
2. Right-click and select **Copy**.
3. Expand **FNDATTCH** and select **Alerts**.
4. Right-click and select **Paste**.

Step 13: Edit the WHEN-BUTTON-PRESSED trigger

1. In the **Object Navigator**, expand **FNDATTCH > Data Blocks > DOCUMENT_CONTROL > Items > OPEN_OTHER_DOCUMENT > Triggers** and select **WHEN-BUTTON-PRESSED**.
2. Right-click and select **PL/SQL Editor**.
3. Replace any existing code with the following code:

```
if MarkView.IsMarkView then
copy(:document_header.file_name, 'GLOBAL.ATCHMT_FILE_NAME');
APPCORE_CUSTOM.EVENT('OPEN-DOCUMENT');
else
ATCHMT_OPEN_OTHER_DOCUMENT.open_document('document_header');
end if;
```

4. **Program > Compile PL/SQL > Incremental** and close the window.

Step 14: Save your changes

1. In the **Object Navigator**, select **File > Save** to save FNDATTCH.
2. Close Oracle Forms Builder.
3. Navigate to `$c_MARKVIEW_TOP/forms/US`.
4. Using Oracle Forms generator, issue the following command to compile the form into an .fmx file, substituting the appropriate userid information.

Note These commands must be run on a single line. If you cut and paste from this guide, remove any line breaks.

Oracle R12:

```
frmcmp userid=apps/apps-pw@connectstring module=$c_MARKVIEW_TOP/forms/US/
FNDATTCH.fmb module_type=FORM compile_all=YES
```

The compiled form is now in the US forms directory at FNDATTCH.fmx.

Note MarkView has encountered problems when compiling forms in some versions of **Forms Builder**. Compile from the command line to safeguard against these problems.

5. In the host operating system, copy FNDATTCH.fmx to `$FND_TOP/forms/US`.
For example, from a UNIX shell using socket mode, run the following command:

```
$ cp $c_MARKVIEW_TOP/forms/US/FNDATTCH.fmx $FND_TOP/forms/US
```

Note You do not need to copy FNDATTCH.fmb to `$FND_TOP/forms/US`. At runtime, the system uses the .fmx file, not the .fmb file.

Update the custom library (CUSTOM.pll)

The custom library (CUSTOM.pll) is an Oracle Forms library shipped with Oracle Applications. The custom library is attached to all forms in the E-Business Suite. As shipped by Oracle, this library performs no functions, but provides callouts for custom logic.

The MarkView installer overwrites CUSTOM.pll to integrate it with MarkView. Therefore, if you added your own customizations to CUSTOM.pll, you must restore the version of CUSTOM.pll that contains your modifications and manually add the MarkView integration code.

Step 1: Back up the active CUSTOM.pll and CUSTOM.plx files

Back up the active versions of the CUSTOM.pll and CUSTOM.plx files before you restore the modified CUSTOM.pll file.

1. Create the following directory (if it does not already exist):

```
$c_MARKVIEW_TOP/MVOA/<version_number>/restore
```

where <version_number> is the Oracle version number.

2. Copy:

```
$AU_TOP/resource/CUSTOM.plx
```

to:

```
$c_MARKVIEW_TOP/MVOA/<version_number>/restore
```

3. Copy:

```
$AU_TOP/resource/CUSTOM.pll
```

to:

```
$c_MARKVIEW_TOP/MVOA/<version_number>/restore
```

Step 2: Restore the modified CUSTOM.pll file

If you did not modify the CUSTOM.pll file, skip this step and proceed to [Step 3: Open CUSTOM.pll for editing](#) on page 25.

If you added your own customizations to CUSTOM.pll, restore the version of the file that contains your changes. The MarkView installer creates restore directories with both CUSTOM.pll and CUSTOM.plx files in

```
$c_MARKVIEW_TOP/MVOA/<version_number>/Restore_<timestamp>
```

where <version_number> is the Oracle version number, and <timestamp> is the date and time when the MarkView installer ran.

1. Locate the original CUSTOM.pll file in `$c_MARKVIEW_TOP/MVOA/<version_number>/Restore_<timestamp>`.

If multiple restore directories exist, use the directory which contains the most recent version of your modified files.

2. Copy:

```
$c_MARKVIEW_TOP/MVOA/<version_number>/Restore_<timestamp>/CUSTOM.pll
```

to:

```
$c_MARKVIEW_TOP/MVOA/<version_number>/libraries/
```

Step 3: Open CUSTOM.pll for editing

1. Start Oracle Forms Builder as described in [Start Oracle Forms Builder](#) on page 20.

2. Connect to the Oracle APPS database schema.
 - a. Select **File > Connect**.
 - b. Enter the Oracle APPS database user, password, and connect string (the same credentials specified during MarkView for Oracle Applications installation).
3. Select **File > Open** and open:
`$c_MARKVIEW_TOP/MVOA/<version_number>/libraries/CUSTOM.pll`
where <version_number> is the Oracle version number.
The name of the library, CUSTOM, appears under PL/SQL Libraries in the navigation tree.

Step 4: Attach libraries (customized versions of CUSTOM.pll only)

If you did not customize CUSTOM.pll, skip this step and proceed to [Step 5: Edit the zoom_available and style functions and the event procedure](#) on page 26. If you customized CUSTOM.pll, manually attach the following MarkView libraries to CUSTOM.pll:

- MVOAUTIL.pll
- MVVIEWER.pll
- MVFOLDER.pll
- SFAPI.pll

1. In the Object Navigator, expand **PL/SQL Libraries > CUSTOM** and select **Attached Libraries**.
2. Click the plus sign **+** button in the toolbar to create a new attached library.
The Attach Library window opens.
3. Click **Browse**.
The **PL/SQL Library File** window opens.
4. Navigate to `$AU_TOP/resource` and select **MVOAUTIL.pll**.
5. Click **OK**.
The **PL/SQL Library File** window closes.
6. Click **Attach**.
7. Repeat these steps for MVVIEWER.pll, MVFOLDER.pll, and SFAPI.pll.

All four attached libraries are visible under Attached Libraries in the tree.

Step 5: Edit the zoom_available and style functions and the event procedure

The following changes modify zoom_available, style, and event so that they call the MVOAUTIL library. Since these modifications require some additional analysis, contact Kofax Technical Support before you make these changes to CUSTOM.pll.

1. In the **Object Navigator**, expand **PL/SQL Libraries > CUSTOM > Program Units** and select **CUSTOM* (Package Body)@**.
2. Right-click and select **PL/SQL Editor**.
3. In the **function zoom_available** section, comment out any existing return lines, such as:

```
# return(FALSE);
```

4. Add the following code to the **function zoom_available** section immediately before the end line:

```
return(MVOA_Util.Zoom_Available);
```

5. In the **function style** section, comment out any existing return lines, such as:

```
# return (FALSE);
```

6. Add the following code to the end of the **function style** section immediately before the end line:

```
return(MVOA_Util.Zoom_Style);
```

7. In the **procedure event** section, comment out any existing return lines, for example:

```
# return (FALSE);
```

8. Add the following code to the end of the **procedure event** section immediately before the end line:

```
MVOA_Util.Event(event_name);
if ((GET_APPLICATION_PROPERTY(Current_Form_Name) = 'APXXXEER'
or GET_APPLICATION_PROPERTY(Current_Form_Name) = 'SFXIISIM'
or GET_APPLICATION_PROPERTY(Current_Form_Name) = 'APXINWKB')
and event_name = 'ZOOM') then
execute_trigger('WHEN-WINDOW-ACTIVATED');
end if;
```

9. Select **File > Save** and close the PL/SQL Editor.

Step 6: Compile and copy CUSTOM.pll

1. Using **Oracle Forms generator**, issue the following command to compile CUSTOM.pll into CUSTOM.plx. In this command, <version_number> is the Oracle version number:

Note These commands must be run on a single line. If you cut and paste from this guide, remove any line breaks and update the version number.

```
frmcmp userid=apps/apps-pw@connectstring module=%c_MARKVIEW_TOP/MVOA/
<version_number>/libraries/CUSTOM.pll module_type=LIBRARY compile_all=YES
```

2. Copy:

```
%c_MARKVIEW_TOP/MVOA/<version_number>/libraries/CUSTOM.pll
```

to:

```
%AU_TOP/resource
```

where <version_number> is the Oracle version number.

3. Copy:

```
%c_MARKVIEW_TOP/MVOA/<version_number>/libraries/CUSTOM.plx
```

to:

```
%AU_TOP/resource
```

where <version_number> is the Oracle version number.

Enable Invoice Workbench (optional)

If you plan to use Oracle forms with the MarkView Process GetNext or Open Work Item operations, you must modify copies of the original Oracle forms. This section describes how to enable Invoice Workbench functionality for MarkView Process workflow processing, by creating a modified version of the Invoice Workbench form (APXINWKB). This modified form (SFXINWKB) will be enabled for both GetNext and Open Work Item.

The following instructions assume that the MarkView base path is the default, %c_MARKVIEW_TOP.

Step 1: Image-enable the default Invoice Workbench form (optional)

There are two versions of the Invoice Workbench form:

- The integrated version (SFXINWKB.fmb). Use this version when you require GetNext and Open Work Item workflow processing from the Invoice Workbench.
- The unintegrated version (APXINWKB.fmb). Use this version when you do not require GetNext or Open Work Item workflow processing from the Invoice Workbench.

If you use the integrated version, skip this step and go to [Step 2: Copy the Invoice Workbench form](#) on page 28.

If you use the unintegrated version, image-enable the default invoice workbench form.

1. On the Oracle E-Business Suite server, open a command shell and log in as the Oracle E-Business suite owner. The default account for this user is applmgr.
2. Navigate to:

```
<distribution directory>\modules\erp-integration-oracle-dist-x.x.x\erp-integration-oracle-db\schema\seed_data\oracle_forms
```

where x.x.x is the MarkView version number.
3. Run APXINWKB.sql.
4. Continue at [Configure Oracle Forms for multi-language support \(optional\)](#) on page 31.

Step 2: Copy the Invoice Workbench form

1. Copy:

```
$AU_TOP/forms/US/APXINWKB.fmb
```

to:

```
$C_MARKVIEW_TOP/forms/US
```
2. Rename:

```
$C_MARKVIEW_TOP/forms/US/APXINWKB.fmb
```

to:

```
$C_MARKVIEW_TOP/forms/US/SFXINWKB.fmb
```

Step 3: Open the SFXINWKB form

1. Start Oracle Forms Builder as described in [Start Oracle Forms Builder](#).
2. Connect to the Oracle APPS database schema.
 - a. Select **File > Connect**.
 - b. Enter the Oracle APPS database user, password, and connect string (the same credentials specified during MarkView for Oracle Applications installation).
3. Select **File > Open** and open `$C_MARKVIEW_TOP/forms/US/SFXINWKB.fmb`.

Note The name of the form, APXINWKB, appears at the top of the navigation tree. Do not change this name to SFXINWKB. Only the .fmb and .fmx file names change.

Step 4: Edit the IMG_IMAGE_ID item

1. In the **Object Navigator**, expand **APXINWKB > Data Blocks > INV_SUM_FOLDER > Items**.
2. Click the plus sign **+** button in the toolbar to create a new item.
The new item appears at the top of the tree.
3. Right-click the item and select **Property Palette**.
4. In the Property Palette **General** section **Name** field, enter **IMG_IMAGE_ID**.
5. In the **Item Type** field, select **Text Item**.
6. In **Subclass Information**, click the button to the right of the blank field.
The Subclass Information window opens.
7. Select **Property Class** and click **OK**.
8. In the Property Palette **Data** section **Maximum Length** field, enter **240**.
9. In the Property Palette **Database** section **Database Item** field, select **No**.
10. Close the Property Palette.

Step 5: Edit the Invoice Folder

1. Expand **APXINWKB > Program Units** and select **INV_SUM_FOLDER_BLOCK_INSERT1* (Package Body)**.
2. Right-click and select **PL/SQL Editor**.
3. Add the following code immediately before the END Insert_Row line:


```
IF :INV_SUM_FOLDER.IMG_IMAGE_ID IS NOT NULL THEN
APPCORE_CUSTOM.EVENT('INVOICE-IMAGE-ASSOCIATED');
END IF;
```
4. **Program > Compile PL/SQL > Incremental** and close the window.

Step 6: Edit the KEY-DUPREC trigger

1. In the Object Navigator, expand **APXINWKB > Data Blocks > INV_SUM_FOLDER > Triggers** and select **KEY-DUPREC**.
2. Right-click and select **PL/SQL Editor**.
3. Replace the existing text with the following:


```
-- inv_sum_folder.event('KEY-DUPREC');
Declare
  cur_image_id Varchar2(240);
Begin
  cur_image_id := :inv_sum_folder.img_image_id;
  inv_sum_folder.event('KEY-DUPREC');
  :inv_sum_folder.img_image_id := cur_image_id;
End;
```
4. **Program > Compile PL/SQL > Incremental** and close the window.

Step 7: Create the WHEN-WINDOW-ACTIVATED trigger

1. In the **Object Navigator**, expand **APXINWKB** and select **Triggers**.

2. Click the plus sign + button in the toolbar to create a new trigger.
A list of trigger names appears.
3. From the list, select **WHEN-WINDOW-ACTIVATED** and click **OK**.
The PL/SQL Editor opens.
4. Add the following code to the new WHEN-WINDOW-ACTIVATED trigger:

```
APPCORE_CUSTOM.EVENT('MVOA-WHEN-WINDOW-ACTIVATED');
```

5. **Program > Compile PL/SQL > Incremental** and close the window.

Step 8: Edit the SFXINWKB form revision number

1. In the **Object Navigator**, expand **APXINWKB > Program Units** and select **APXINWKB (Package Body)**.
2. Right-click and select **PL/SQL Editor**.
3. Append **.170** to the revision number to indicate that the form is edited for MarkView integration.
For example, change:
FND_STANDARD.FORM_INFO('\$Revision: 115.155.11591.2')
to:
FND_STANDARD.FORM_INFO('\$Revision: 115.155.11591.2.170')
4. **Program > Compile PL/SQL > Incremental** and close the window.

Step 9: Compile the Invoice Workbench form

1. In the **Object Navigator**, select **File > Save** to save SFXINWKB.
2. Close Oracle Forms Builder.
3. Navigate to `%c_MARKVIEW_TOP/forms/US`.
4. Using Oracle Forms generator, issue the correct command to compile the form into an .fmx file, substituting the appropriate userid information.

Note These commands must be run on a single line. If you cut and paste from this guide, remove any line breaks.

```
frmcmp userid=apps/apps-pw@connectstring module=%c_MARKVIEW_TOP/forms/US/SFXINWKB.fmb module_type=FORM compile_all=YES
```

The compiled form is now at `%c_MARKVIEW_TOP/forms/US/SFXINWKB.fmx`.

Note MarkView has encountered problems when compiling forms in some versions of **Forms Builder**. Compile from the command line to safeguard against these problems.

Configure Oracle Forms for multi-language support (optional)

To configure Oracle Forms to support languages in addition to the default, US English, edit language-specific versions of the forms to include the MarkView integration code.

1. Locate the FNDATTCH.fmb file for the target language. For example, the US English version of this file is located at:

```
$AU_TOP/forms/US/FNDATTCH.fmb
```

and the French version of the file is located at:

```
$AU_TOP/forms/F/FNDATTCH.fmb
```

2. Create a new language-specific subdirectory under `$c_MARKVIEW_TOP/forms`. For example, if you are adding support for French, create:

```
$c_MARKVIEW_TOP/forms/F
```

3. Modify the FNDATTCH.fmb file for the target language to include the MarkView integration code, as described in [Update the Oracle Attachments form \(FNDATTCH.fmb\)](#) on page 20.
4. Copy the APXINWKB.fmb file from the language-specific subdirectory in `$AU_TOP/forms` to the new language-specific subdirectory under `$c_MARKVIEW_TOP/forms` that you created in step 2.
5. Rename the APXINWKB.fmb file to SFXINWKB.fmb.
6. Modify the SFXINWKB.fmb file for the target language to include the MarkView integration changes, as described in the *Kofax MarkView Installation Guide*.
7. Use the Oracle Forms Generator to compile SFXINWKB.fmb into SFXINWKB.fmx. Make sure the .fmx file is in the correct language-specific subdirectory.
8. Repeat these steps for each language that you want to support.

Enable MarkView Quick Invoices (optional)

If Kofax Transformation Modules for MarkView is a part of your solution, enable MarkView Quick Invoices.

Step 1: Copy the Quick Invoices form

1. Copy:

```
$AU_TOP/forms/US/APXIISIM.fmb
```

to:

```
$c_MARKVIEW_TOP/forms/US
```

2. Rename:

```
$c_MARKVIEW_TOP/forms/US/APXIISIM.fmb
```

to:

```
$c_MARKVIEW_TOP/forms/US/SFXIISIM.fmb
```

Step 2: Open the SFXIISIM form

1. Start Oracle Forms Builder as described in [Start Oracle Forms Builder](#).

2. Connect to the Oracle APPS database schema.
 - a. Select **File > Connect**.
 - b. Enter the Oracle APPS database user, password, and connect string (the same credentials specified during MarkView for Oracle Applications installation).
3. Select **File > Open** and open `$_c_MARKVIEW_TOP/forms/US/SFXIISIM`.
4. In the **Property Palette**, update the form name from APXIISIM to SFXIISIM at the top of the tree.

Step 3: Edit the IMG_IMAGE_ID item

1. In the **Object Navigator**, expand **SFXIISIM > Data Blocks > INVOICES_FOLDER > Items**.
2. Click the plus sign **+** button in the toolbar to create a new item.
The new item appears at the top of the tree.
3. Right-click the item and select **Property Palette**.
4. In the Property Palette **General** section **Name** field, enter **IMG_IMAGE_ID**.
5. In the **Item Type** field, select **Text Item**.
6. In **Subclass Information**, click the button to the right of the blank field.
The Subclass Information window opens.
7. Select **Property Class** and click **OK**.
The Subclass Information window closes.
8. In the Property Palette **Data** section **Maximum Length** field, enter **240**.
9. In the Property Palette **Database** section **Database Item** field, select **No**.
10. Close the Property Palette.

Step 4: Edit the ON-INSERT trigger

1. In the **Object Navigator**, expand **SFXIISIM > Data Blocks > INVOICES_FOLDER > Triggers** and select **ON-INSERT**.
2. Right-click and select **PL/SQL Editor**.
3. Add the following code after the call to `INVOICES_FOLDER.INSERT_ROW`:

```
IF :INVOICES_FOLDER.IMG_IMAGE_ID IS NOT NULL THEN
  APPCORE_CUSTOM.EVENT('INTERFACE-INVOICE-IMAGE-ASSOCIATED');
END IF;
```

4. **Program > Compile PL/SQL > Incremental** and close the window.

Step 5: Edit the KEY-DUPREC trigger

1. In the **Object Navigator**, expand **SFXIISIM > Data Blocks > INVOICES_FOLDER > Triggers** and select **KEY-DUPREC**.
2. Right-click and select **PL/SQL Editor**.
3. Replace the existing text with the following:

```
--invoices_folder.key_duprec;
Declare
  cur_image_id Varchar2(240);
Begin
  cur_image_id:=:invoices_folder.img_image_id;
  invoices_folder.key_duprec;
```

```
:invoices_folder.img_image_id:=cur_image_id;
End;
```

4. **Program > Compile PL/SQL > Incremental** and close the window.

Step 6: Create the WHEN-WINDOW-ACTIVATED trigger

1. In the **Object Navigator**, expand **SFXIISIM** and select **Triggers**.
2. Click the plus sign **+** button in the toolbar to create a new trigger.
A list of trigger names appears.
3. From the list, select **WHEN-WINDOW-ACTIVATED** and click **OK**.
The PL/SQL Editor opens.
4. Replace the existing code with:

```
default_value(null, 'global.mvoa_open_opening_work_item');
if :GLOBAL.MVOA_OPEN_OPENING_WORK_ITEM = 'Y' then
  clear_block(no_validate);
end if;
APPCORE_CUSTOM.EVENT('MVOA-WHEN-WINDOW-ACTIVATED');
```

5. **Program > Compile PL/SQL > Incremental** and close the window.

Step 7: Edit the WHEN-NEW-FORM-INSTANCE trigger

1. In the **Object Navigator**, expand **SFXIISIM > Triggers** and select **WHEN-NEW-FORM-INSTANCE**.
2. Right-click and select **PL/SQL Editor**.
3. Comment out the following line:

```
--EXECUTE_TRIGGER('QUERY_FIND');
```

4. Add the following line:

```
set_window_property('INVOICES_FOLDER_WINDOW', title, 'MarkView Quick Invoices');
```

5. **Program > Compile PL/SQL > Incremental** and close the window.

Step 8: Edit the SOURCE_DSP item

1. In the **Object Navigator**, expand **SFXIISIM > Data Blocks > QF_INVOICES > Items** and select **SOURCE_DSP**.
2. Right-click the item and select **Property Palette**.
3. In the **Data** section, set **Required** to **No**.
4. Close the Property Palette.

Step 9: Edit the GROUP_ID item

1. In the **Object Navigator**, expand **SFXIISIM > Data Blocks > QF_INVOICES > Items** and select **GROUP_ID**.
2. Right-click the item and select **Property Palette**.
3. In the **Data** section, set **Required** to **No**.
4. Close the Property Palette.

Step 10: Edit the Query Data Source Name

1. In the Object Navigator, expand **SFXIISIM > Data Blocks** and select **INVOICES_FOLDER**.
2. Right-click and select **Property Palette**.
3. In the **Database** section, set **Query Data Source Name** to **MVCN_AP_INVOICES_INTERFACE_V**.
4. Close the Property Palette.

Step 11: Edit PRE_UPDATE, LOCK_ROW, and UPDATE_ROW procedures for Oracle EBS 12.x

Oracle enhancements to APXIIISIM and party_site_id in now enable you to create two invoices with the same invoice number, vendor, and vendor_site_id.

If you applied the Oracle patches, edit the PRE_UPDATE, LOCK_ROW, and UPDATE_ROW procedures to configure MarkView to support the enhancements. If you are not sure whether or not you applied the patches, follow these instructions; if the text of your PRE_UPDATE, LOCK_ROW, and UPDATE_ROW procedures do not match those in a particular step, skip that step.

1. Expand **SFXIISIM > Program Units**.
2. Right-click **INVOICES_FOLDER_EVENT (Package Body)** and select **PL/SQL Editor**.
3. Edit the PRE_UPDATE procedure:
 - a. In the PRE_UPDATE procedure, locate the ap_invoices_interface section:

```
select count(1)
into x_cnt_inv_int
from ap_invoices_interface
where invoice_num = :invoices_folder.invoice_num
and vendor_id = :invoices_folder.vendor_id
and (party_site_id = :invoices_folder.party_site_id
/*Bug9105666*/
OR (party_site_id is null and :invoices_folder.party_site_id is
null)) /*Bug9105666*/
```

Note If the PRE_UPDATE procedure does not include these lines, continue at step 4.

- b. Edit these lines:

```
and (party_site_id = :invoices_folder.party_site_id
/*Bug9105666*/
OR (party_site_id is null and :invoices_folder.party_site_id is
null)) /*Bug9105666*/
```

- To allow the same invoice number for another vendor_site_id, replace those lines with:

```
and (vendor_site_id = :invoices_folder.vendor_site_id /*Bug9105666*/
OR (vendor_site_id is null and :invoices_folder.vendor_site_id is
null)) /*Bug9105666*/
```

- To prevent the same invoice number from being used twice for the same vendor, comment out the lines.

4. Click **Program > Compile PL/SQL > Incremental** and close the window.
5. Right-click **INVOICES_FOLDER_EVENT_LOCK (Package Body)** and select **PL/SQL Editor**.

6. Edit the **INVOICES_FOLDER_EVENT_LOCK.LOCK_ROW** procedure:

- a. Locate the LOCK_ROW procedure.
- b. If the following line appears in the LOCK_ROW procedure, comment out the line:

```
x_party_site_id => :invoices_folder.party_site_id
/*Bug 16361548*/
```

7. Click **Program > Compile PL/SQL > Incremental** and close the window.
8. Right-click **INVOICES_FOLDER_EVENT_UPDATE (Package Body)** and select **PL/SQL Editor**.
9. Edit the **INVOICES_FOLDER_EVENT_UPDATE.UPDATE_ROW** procedure:

- a. Locate the UPDATE_ROW procedure.
- b. If the following line appears in the UPDATE_ROW procedure, comment out the line:

```
x_party_site_id => :invoices_folder.party_site_id
/*Bug 16361548*/
```

10. Click **Program > Compile PL/SQL > Incremental** and close the window.

Step 12: Edit INVOICES_CONTENT

1. In the **Object Navigator**, expand **SFXIISIM > Canvases** and select **INVOICES_CONTENT**.
2. Right-click and select **Layout Editor**.
3. Right-click **Save and Next** in the lower right corner of the form to open the **Property Palette**.
4. In the **Functional** section, set **Enabled** to **No**.
5. In the **Physical** section, set **Visible** to **No**.
6. Close the **Property Palette**.
7. Close **INVOICES_CONTENT**.

Step 13: Save your changes

1. In the **Object Navigator**, select **File > Save** to save SFXIISIM.
2. Close Oracle Forms Builder.
3. Navigate to `c_MARKVIEW_TOP/forms/US`.
4. Using Oracle Forms generator, issue the following command to compile the form into an .fmx file, substituting the appropriate userid information.

Note These commands must be run on a single line. If you cut and paste from this guide, remove any line breaks.

```
frmcmp userid=apps/apps-pw@connectstring module=$c_MARKVIEW_TOP/forms/US/
SFXIISIM.fmb module_type=FORM compile_all=YES
```

The compiled form is now at `$c_MARKVIEW_TOP/forms/US/SFXIISIM.fmx`.

Note MarkView has encountered problems when compiling forms in some versions of **Forms Builder**. Compile from the command line to safeguard against these problems.

Remove MarkView personalizations from Supplier form (optional)

When you install the MarkView Oracle objects, MarkView personalizes the Supplier form. To remove the personalizations:

1. Log in to Oracle Applications as the Functional Administrator.
2. Navigate to **Personalizations > Import/Export**.
3. In the **Personalization Repository** window, navigate to `/oracle/apps/pos/supplier/webui/customizations/site/0/`.
4. Select each of the following site-level customizations and click **Delete**:
 - OrganizationPG.xml
 - QuickUpdatePG.xml
 - SuppCrtPG.xml
 - SuppSummPG.xml

Synchronize the run and patch file systems (Oracle EBS 12.2 only)

Follow these steps if you are integrating MarkView with Oracle E-Business Suite 12.2 or higher.

Proceed to one of the following procedures to synchronize the MarkView files between the run and patch file systems:

- [Synchronize the MarkView files using adop phase=fs_clone command](#)

The file systems synchronization with `adop phase=fs_clone` command requires a large amount of free disk space to be available for `adop` operations. The synchronization procedure is time-consuming. For more information about space requirements, refer to the Oracle documentation.
- [Synchronize the MarkView files using custom synchronization driver \(for UNIX only\)](#)

This procedure requires less time and disk space compared to the synchronization using `adop phase=fs_clone`.

Synchronize the MarkView files using adop phase=fs_clone command

If you configured Oracle Forms or applied any changes to the Oracle Forms, do the following:

1. Before synchronizing the run and patch file systems, verify that no patching cycle is currently active. For example, from a UNIX shell, run the following command:

```
$ adop -status
```

2. To propagate the changes to the secondary file system, run the following command:

```
$ adop phase=fs_clone
```

`adop phase=fs_clone` command recreates the patch edition file system as an exact copy of the run edition file system.

3. For more information about using the Oracle `adop` (the AD Online Patching) utility, refer to the Oracle `adop` utility documentation.

Synchronize the MarkView files using custom synchronization driver (for UNIX only)

If you configured Oracle Forms or applied any changes to the Oracle Forms, do the following:

1. Before synchronizing the run and patch file systems, verify that no patching cycle is currently active. For example, from a UNIX shell, run the following command:

```
$ adop -status
```

2. In `<distribution_directory>\installer\tools`, locate the `copy_mv_objects.sh` shell script, where `<distribution_directory>` is the location where you downloaded and decompressed the installation files.
3. Copy `copy_mv_objects.sh` to: `$APPL_TOP_NE/ad/custom/`
Verify that the copied script has the **Execute** permissions.
4. In `$APPL_TOP_NE/ad/custom/adop_sync.drv`, add the following line between `#Begin Customization` and `#End Customization`:

```
sh %s_ne_base%/EBSapps/appl/ad/custom/copy_mv_objects.sh  
%s_current_base% %s_other_base%
```

Next time you start a new patching cycle with the `adop phase=prepare` command, the MarkView files will be synchronized between the file systems.

Note The files will be automatically synchronized between the file systems each time you start a new patching cycle with the `adop phase=prepare` command.

If you do not want to propagate any subsequent changes in the MarkView files (for example, temporary changes in the customized Oracle Forms) from the run file system to the patch file system, comment out the added line in `$APPL_TOP_NE/ad/custom/adop_sync.drv` to stop synchronization.

Chapter 4

Expense Management Configuration

MarkView Expense Management is an optional MarkView product. If a user enters an expense report that requires receipts, MarkView temporarily stores the current Oracle expense report status in a configurable field and assigns a MarkView status of "Awaiting Receipts". After the user submits and attaches the required receipts to the expense report, MarkView changes the report status to the original status and releases the expense report to continue through the Oracle workflow.

MarkView Expense Management uses a system-level preference, `EXPENSE_REPORT_STATUS_BACKUP_ATTRIBUTE`, to store the value of the configurable field in the Oracle iExpense table `ap_expense_report_headers_all`. Valid values are `ATTRIBUTE1` through `ATTRIBUTE15`. The default value is `ATTRIBUTE15`. Change the default value of this preference if you already use `ATTRIBUTE15` for other purposes.

Configure Oracle Expenses workflow

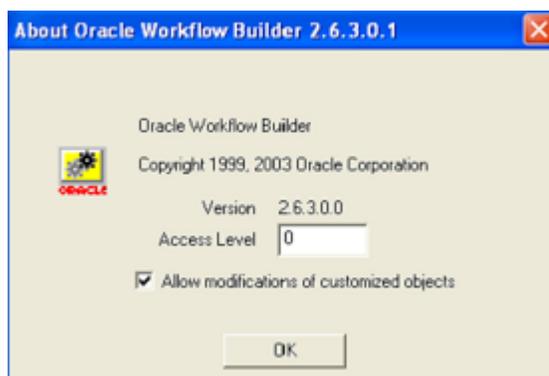
To configure MarkView for Expense Management, you must have Oracle Workflow Builder installed and your system must include data.

The windows in this chapter may differ from the windows on your Oracle user interface. The appearance depends on the version of Oracle Internet Expenses to which you are integrating. Regardless of the appearance, the integration steps are identical.

Set the administrator access level

1. Open **Oracle Workflow Builder**.
2. In **Oracle Workflow Builder**, select **Help > About Oracle Workflow Builder**.

The **About Oracle Workflow Builder** window appears.

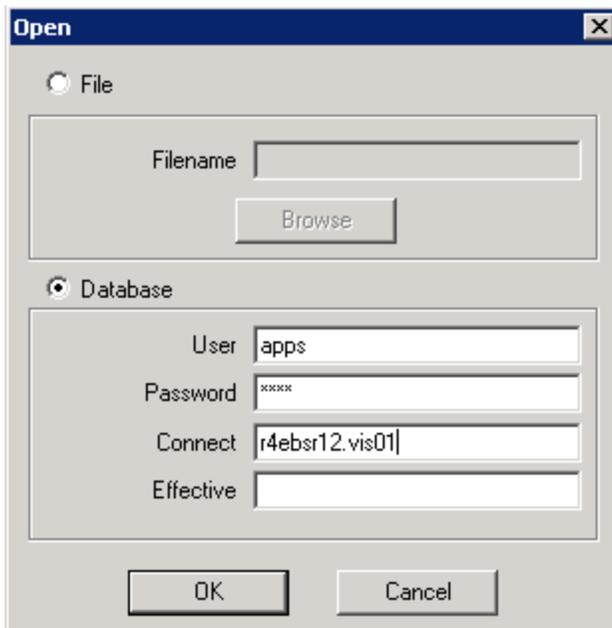


3. Set the values in the **About Oracle Workflow Builder** window:
 - Set **Access Level** to **0**.
 - Select **Allow modification of customized objects**.
4. Click **OK**.

Open the Expenses workflow

1. Start **Oracle Workflow Builder**.
2. Select **File > Open**.

The **Open** window appears.

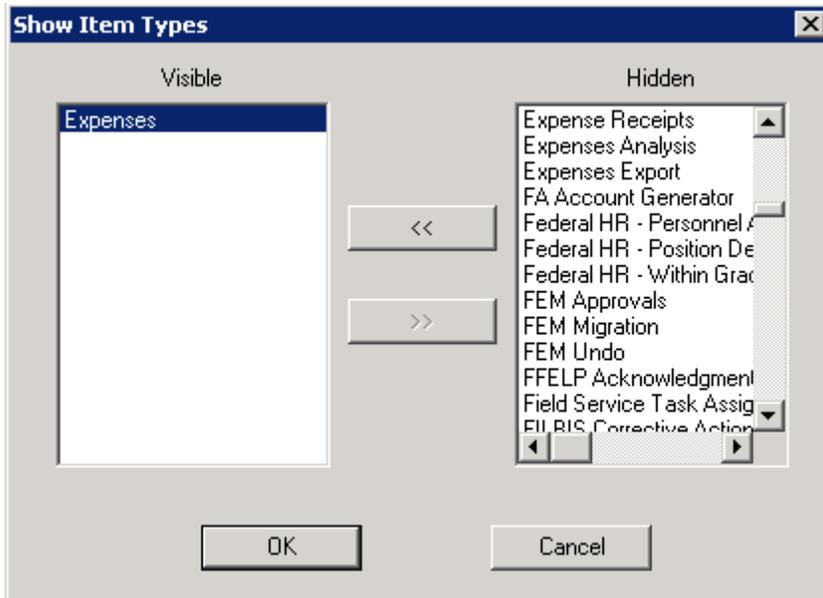


The screenshot shows the 'Open' dialog box with the following fields and values:

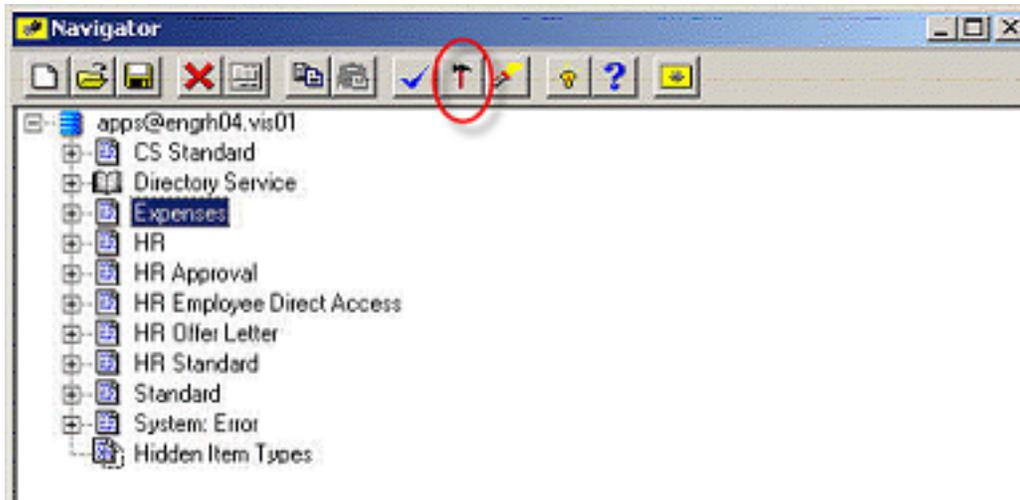
- File** (radio button): Unselected
- Database** (radio button): Selected
- Filename**: Empty text box
- Browse**: Button
- User**: apps
- Password**: ****
- Connect**: r4ebsr12.vis01
- Effective**: Empty text box
- OK**: Button
- Cancel**: Button

3. Enter the following information in the **Open** window:
 - **Select Database**.
 - **User**: Enter the database user name.
 - **Password**: Enter the password associated with the database user name.
 - **Connect**: Enter the connect string information.
 - **Effective**: Leave blank.

4. Click **OK**.
The **Show Item Types** window opens.



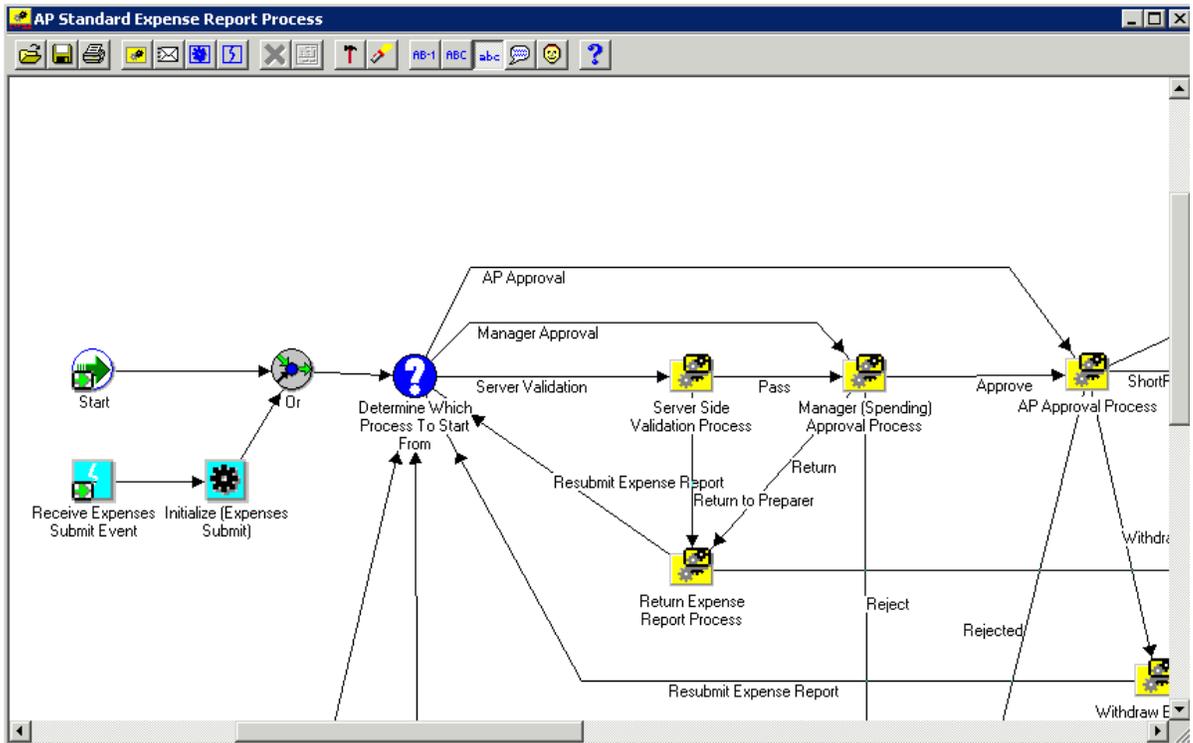
5. Move **Expenses** from the **Hidden** column to the **Visible** column and click **OK**.
The expenses item type appears in a tree structure.
6. Expand the tree to display all the options under the **Expenses** menu tree. The Expenses workflow is now visible.
7. Ensure that the Developer mode is off in Oracle Workflow Builder. Clear the hammer icon in the button bar.



Add functions to the workflow process

1. Open **Oracle Workflow Builder**.

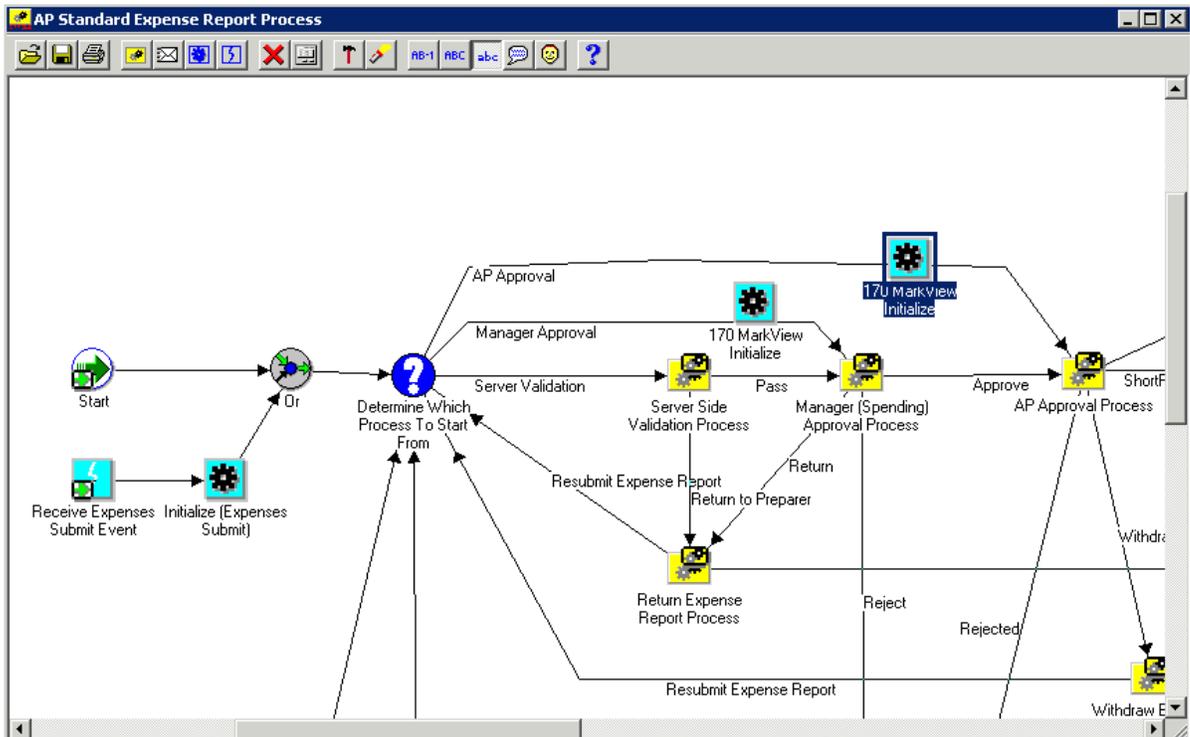
2. Log in with the Oracle Applications database user name and password.
 3. In the **Expenses** item type, expand the **Processes** section.
 4. Double-click **AP Standard Expense Report Process**.
- The main workflow opens in an edit window and the starting process diagram appears.



- From the **Navigator** window, select and drag the MarkView Process Initialize function onto the AP Approval transition.

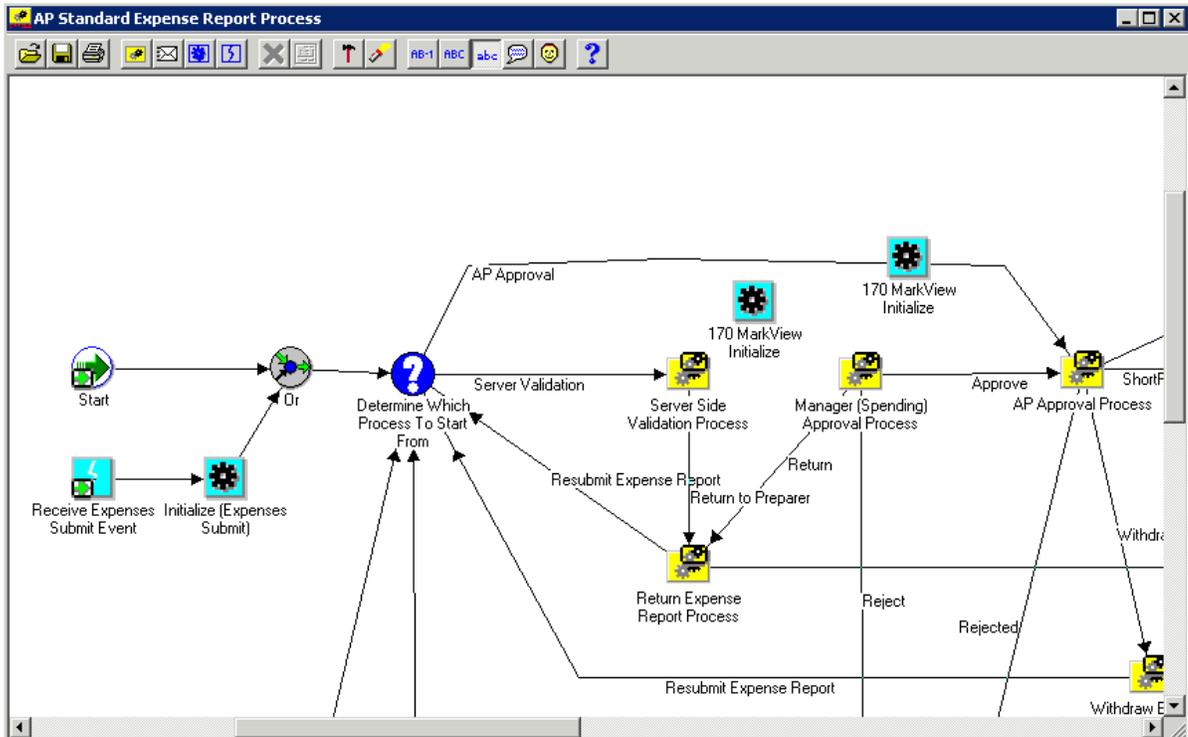
The process diagram contains two instances of MarkView Process Initialize.

Note The instances of MarkView Process Initialize may be labeled SQLFLOW_INIT.



- Delete the **Manager Approval** transition between **Determine Which Process To Start From** and **Manager (Spending) Approval Process**. Right-click the arrow and select **Delete Selection**.

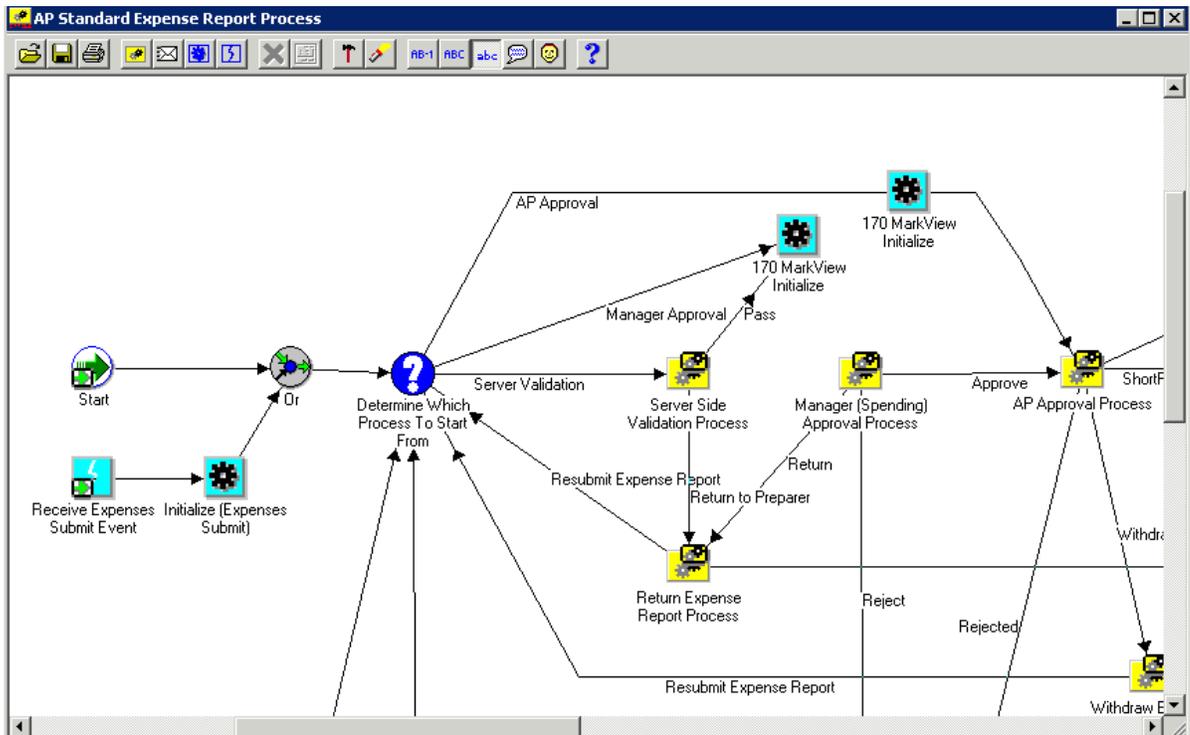
8. Delete the **Pass** transition line between **Server Side Validation Process** and **Manager (Spending) Approval Process**. Right-click the arrow and select **Delete Selection**.



9. Create a transition between the **Determine Which Process To Start From** and **MarkView Process Initialize**:
 - Right-click and hold **Determine Which Process To Start From**.
 - Drag the cursor over **MarkView Process Initialize**.
 - Release the mouse button. A menu appears.
 - Select **Manager Approval**.

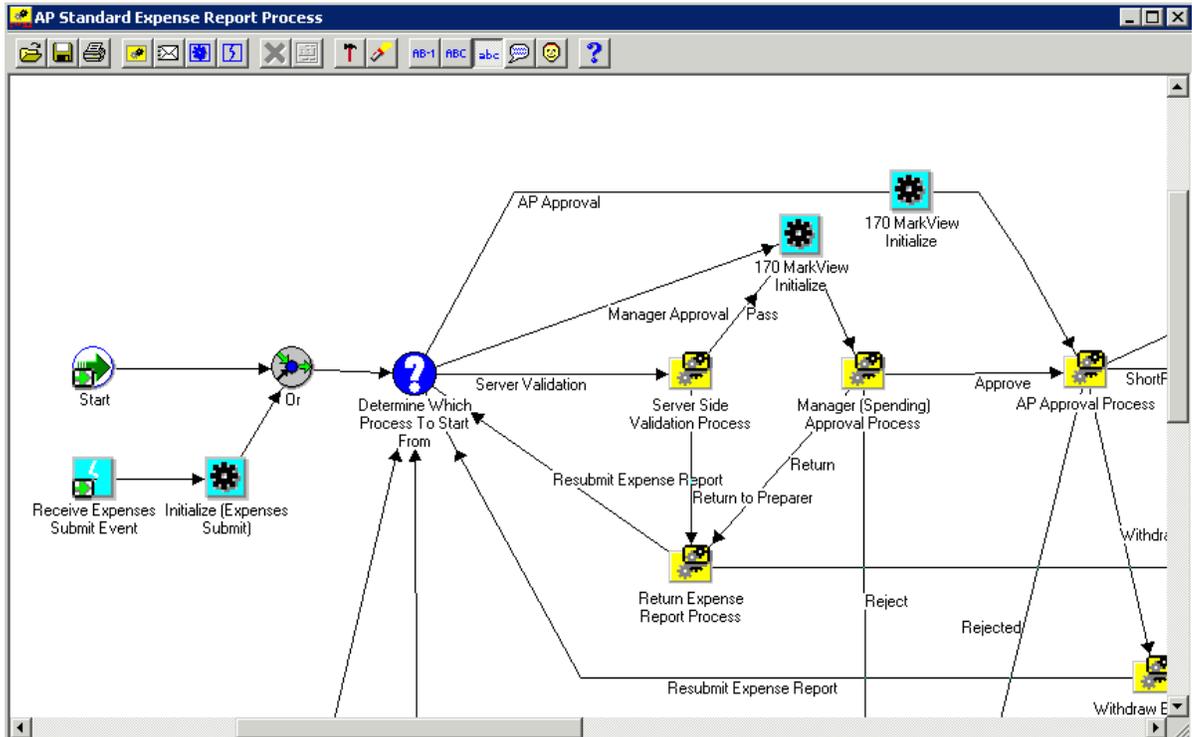
10. Create a transition between **Server Side Validation Process** and **MarkView Process Initialize**:

- Right-click and hold **Server Side Validation Process**.
- Drag the cursor over **MarkView Process Initialize**.
- Release the mouse button. A menu appears.
- Select **Pass**.



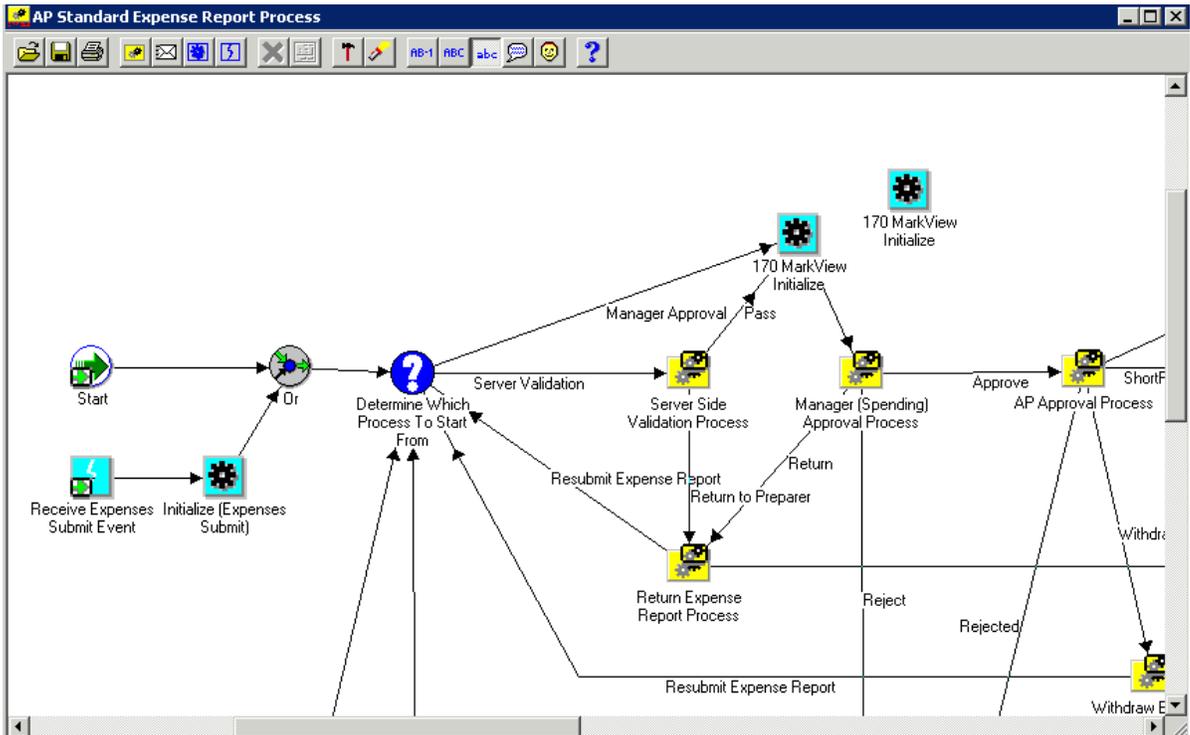
11. Create a transition between **MarkView Process Initialize** and **Manager Approval Process**:

- Right-click and hold **MarkView Process Initialize**.
- Drag the cursor over **Manager Approval Process**.
- Release the mouse button.



12. Delete the AP Approval transition line between Determine Which Process To Start From and AP Approval Process:

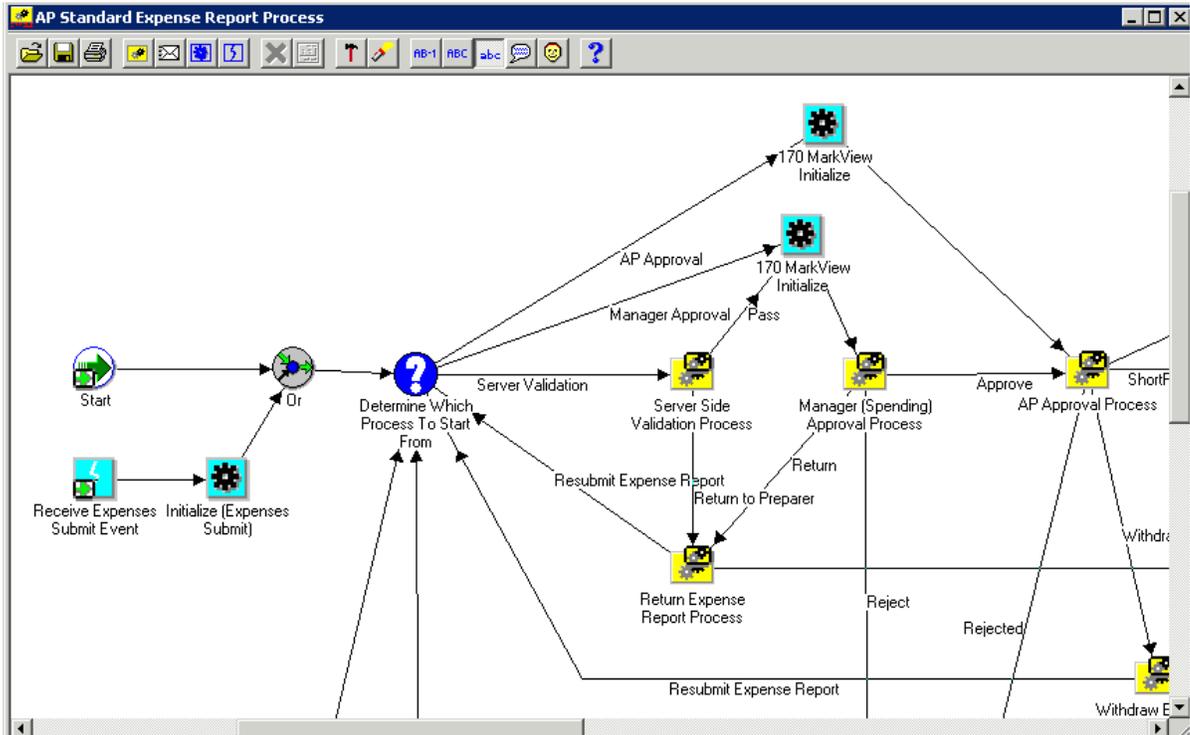
- Position the cursor on the arrow that connects them.
- Right-click and select **Delete Selection**.



13. Create a transition between Determine Which Process to Start From and the second MarkView Process Initialize function:

- a. Right-click and hold **Determine Which Process to Start From**.
- b. Drag the cursor over MarkView Process Initialize.
- c. Release the mouse button. A menu appears.
- d. Select **AP Approval**.

14. Create a transition between MarkView Process Initialize and AP Approval Process:
 - a. Right-click and hold **MarkView Process Initialize**.
 - b. Drag the cursor over **AP Approval**.
 - c. Release the mouse button.



15. Click **Save**. Saving your changes can generate multiple validation failed for activity errors such as:

'SAVE validation failed for activity 'HRSSA/HR_SAVE_FOR_LATER'.

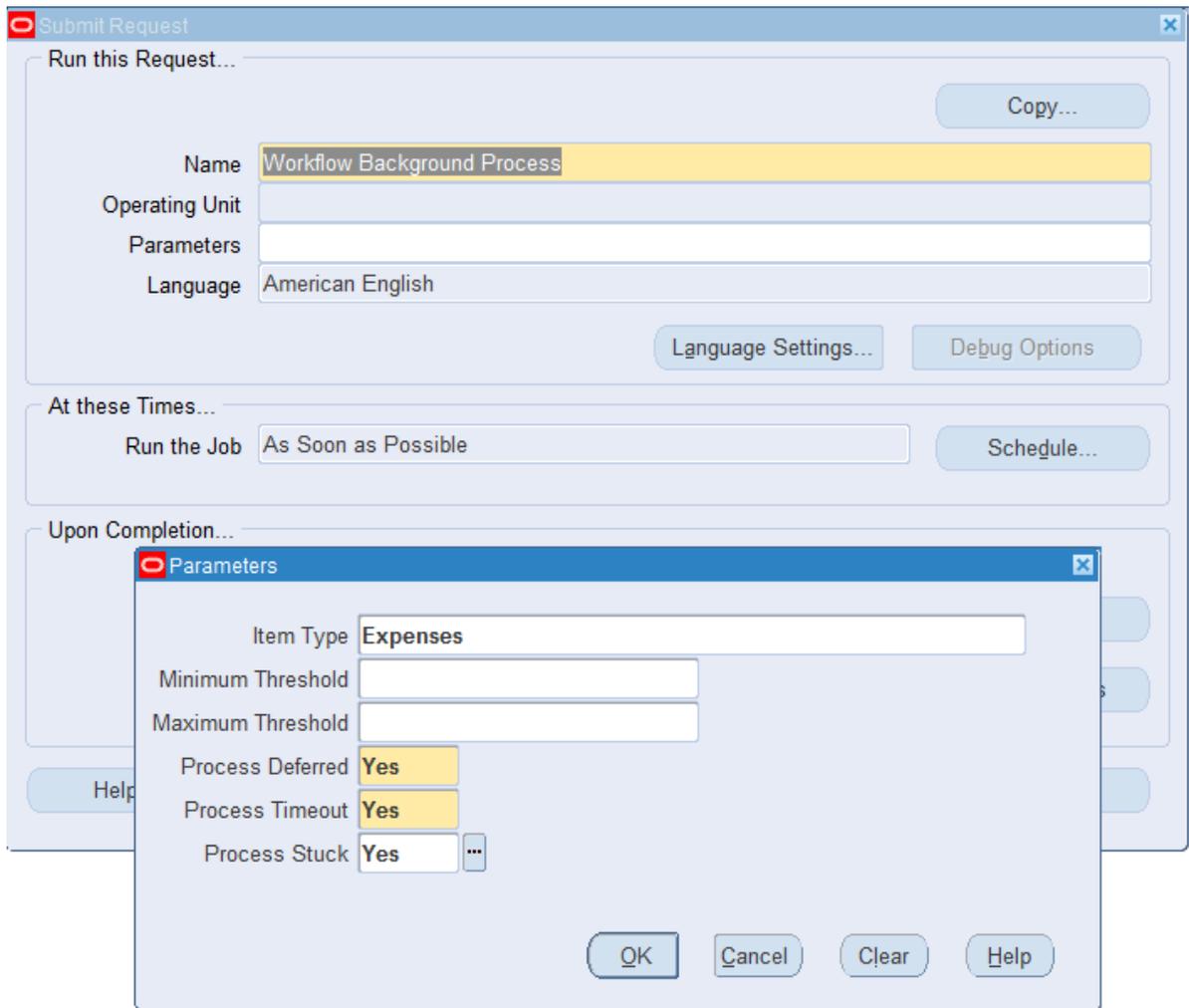
Ignore errors that do not relate to your changes.

Configure the workflow background process

Schedule the Workflow Background Process concurrent request to run periodically in Oracle Concurrent Manager. This concurrent request handles items in the Oracle workflow that are timed out, stuck, or notified. The following steps describe the settings required for use with your MarkView system. For information about setting up this process, see Oracle MetaLink note 182936.1, the Oracle Workflow Developer's Guide, and the Oracle Applications User's Guide.

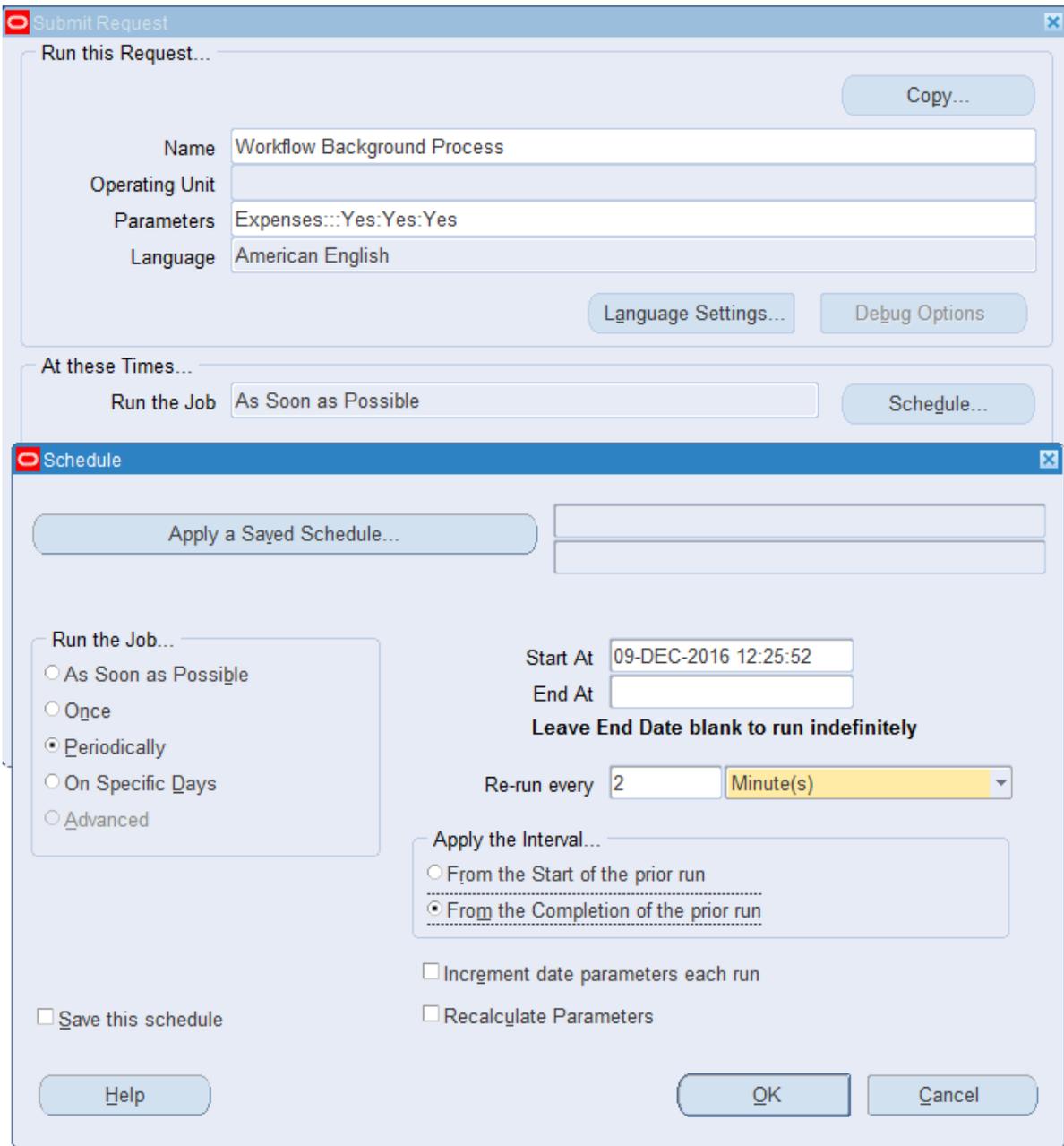
1. Log in to **Oracle Concurrent Manager** as a system administrator.
2. Follow the instructions in MetaLink note 182936.1 and the Oracle Applications User's Guide for submitting the **Workflow Background Process** as a concurrent program.

3. When the **Parameters** window appears, enter the following information.
 - **Item Type:** Expenses
 - **Minimum Threshold:** Leave blank
 - **Maximum Threshold:** Leave blank
 - **Process Deferred:** Yes
 - **Process Timeout:** Yes
 - **Process Stuck:** Yes
4. Click **OK**.
5. Click **Schedule**.



The **Schedule** window appears.

6. Schedule the job to run at 2-minute intervals.



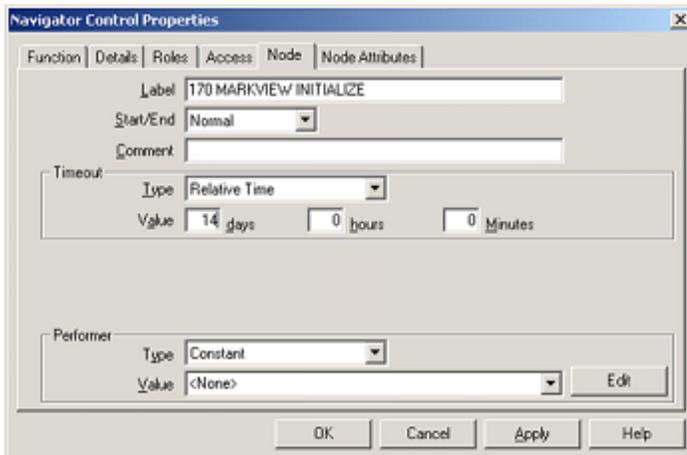
Configure the missing-receipt timeout

When the MarkView Initialize function starts, the system creates the MarkView work item and waits for receipts.

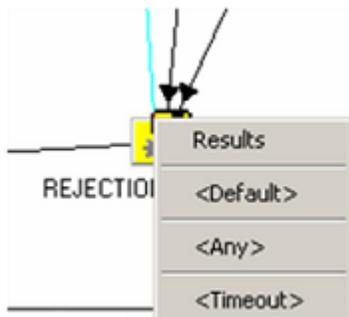
If employees take too long to submit receipts, you can automatically move their expense reports into the Oracle Applications rejection process by configuring the system to reject expense reports after a specified number of days.

To configure the missing-receipt timeout:

1. Open **Oracle Workflow Builder** and log in as the application system user.
2. Open **Expenses (APEXP) Item Type**.
3. Expand the **Processes** list, and double-click the **AP Standard Expense Report** process. The process diagram opens.
4. Right-click the MarkView Initialize function and select **Properties**. The system displays the **Navigator Control Properties** window.
5. Select the **Node** tab.
6. In the **Timeout** section of the tab, change **Type** to **Relative Time**.
7. In the **Value** fields, enter the total time after which an expense report should time out. Enter whole numbers in the **Value** fields.



8. Right-click the **MarkView Initialize** function and draw a line to the REJECTION_PROCESS function. A menu appears.



9. Select **<Timeout>** and save your changes.
10. Repeat this procedure for all other functions that have names similar to MarkView Initialize, such as MarkView Initialize-1.

For this timeout subprocess to work, you must schedule the Workflow Background Process job to execute on a regular basis. A system integrator can configure the Workflow Background process.

Chapter 5

Oracle configuration parameters

Enable Tax Classification Codes

To use the Release 11i model (Tax Classification Codes) in Oracle EBS R12 or 12.2, generate a new ACD configuration using `Use11iTaxMode = TRUE`.

By default, the ACD uses the standard Regime to Rate Flow.

Example:

```
declare
  dfm_definition_data MVERP_DFM_Util.DFMDefinitionData_R;
  dfm_fields          MVERP_DFM_Generate.GenerateFields_T;
begin
  -- extract current DFM data
  MVERP_DFM_Generate.GetToolDfmDefinition(
    ToolName => 'Distributions',
    DFMDefinitionData => dfm_definition_data,
    dfmfields => dfm_fields);

  -- set new values
  dfm_definition_data.Use11iTaxMode := TRUE;

  -- save DFM data
  MVERP_DFM_Generate.GenerateExpressMarkup(
    ToolName => 'Distributions',
    DFMDefinitionData => dfm_definition_data,
    DFMfields => dfm_fields);
end;
/
```

Add the Automatic Distributions check box

Configure Accounting Details to display a check box that instructs MarkView to automatically create distributions based on the accounting line data being entered.

To display the check box, generate a new ACD configuration using `ShowAutoGenerateDistOption = TRUE`.

By default, this check box does not appear.

Example:

```

declare
  dfm_definition_data MVERP_DFM_Util.DFMDefinitionData_R;
  dfm_fields          MVERP_DFM_Generate.GenerateFields_T;

begin
  -- extract current DFM data
  MVERP_DFM_Generate.GetToolDfmDefinition(
    ToolName => 'Distributions',
    DFMDefinitionData => dfm_definition_data,
    dfmfields => dfm_fields);

  -- set new values
  dfm_definition_data.ShowAutoGenerateDistOption := TRUE;

  -- save DFM data
  MVERP_DFM_Generate.GenerateExpressMarkup(
    ToolName => 'Distributions',
    DFMDefinitionData => dfm_definition_data,
    DFMfields => dfm_fields);

end;
/

```

Set default values for the Automatic Distributions check box

The Accounting Details (ACD) tool in MarkView lets you supply coding information for invoices during the invoice life cycle. Oracle R12 introduced the concept of invoice lines and invoice distributions. A line can have multiple distributions. However, many customers maintain a 1:1 relationship between invoice lines and distributions. MarkView provides a check box on the window that you can enable when adding lines to create a single distribution for the line automatically. By default, this check box is disabled.

MarkView includes a preference that lets you change the default setting (disabled) by updating the definition of the ACD tool.

To control the default status of the check box, set ShowAutoGenerateDistOption to **TRUE** and **AutoGenerateDistDefValue** to one of the following:

- TRUE: Enables the check box by default
- FALSE: Disables the check box by default
- NULL: Disables the check box by default

Example:

```

declare
  dfm_definition_data MVERP_DFM_Util.DFMDefinitionData_R;
  dfm_fields          MVERP_DFM_Generate.GenerateFields_T;

begin
  -- extract current DFM data
  MVERP_DFM_Generate.GetToolDfmDefinition(
    ToolName => 'Distributions',
    DFMDefinitionData => dfm_definition_data,
    DFMfields => dfm_fields);

  -- set new values
  dfm_definition_data.ShowAutoGenerateDistOption := TRUE;
  dfm_definition_data.AutoGenerateDistDefValue := TRUE;

end;
/

```

```

-- save DFM data
MVERP_DFM_Generate.GenerateExpressMarkup(
  ToolName => 'Distributions',
  DFMDefinitionData => dfm_definition_data,
  DFMFields => dfm_fields);
end;
/

```

Set default values for Tax fields

Use the Accounting Details (ACD) tool in MarkView to process the Line Type selection and to default the Tax fields. If you enable the Line Type Defaults option, the SetLineTypeDefaultsForLine or SetLineTypeDefaults custom callouts return the default values for the Tax fields after you select Line Type in the Accounting Details form. For more information, see the MarkView Custom Packages in the *Kofax MarkView Administrator's Guide, Volume 2*.

Example:

```

declare
  dfm_definition_data MVERP_DFM_Util.DFMDefinitionData_R;
  dfm_fields          MVERP_DFM_Generate.GenerateFields_T;
begin
  -- extract current DFM data
  MVERP_DFM_Generate.GetToolDfmDefinition(
    ToolName => 'Distributions',
    DFMDefinitionData => dfm_definition_data,
    dfmfields => dfm_fields);

  -- set new values
  dfm_definition_data.EnableLineTypeDefaults := TRUE;

  -- save DFM data
  MVERP_DFM_Generate.GenerateExpressMarkup(
    ToolName => 'Distributions',
    DFMDefinitionData => dfm_definition_data,
    DFMfields => dfm_fields);
end;
/

```

Set maximum values for frequently used accounts

The account information in MarkView Viewer is retrieved from the ERP system, which may be time-consuming. Use the FrequentAccountsLOVSize and FrequentAccountsHistorySize parameters to set the maximum length of the list of values to be displayed in MarkView Viewer and the maximum number of accounts that a user can save in the database. By default, both parameters are set to 50.

Example:

```

declare
  dfm_definition_data MVERP_DFM_Util.DFMDefinitionData_R;
  dfm_fields          MVERP_DFM_Generate.GenerateFields_T;
begin

```

```

-- extract current DFM data
MVERP_DFM_Generate.GetToolDfmDefinition(
    ToolName => 'Distributions',
    DFMDefinitionData => dfm_definition_data,
    dfmfields => dfm_fields);

-- set new values
dfm_definition_data.FrequentAccountsLOVSize := 100;
dfm_definition_data.FrequentAccountsHistorySize := 150;

-- save DFM data
MVERP_DFM_Generate.GenerateExpressMarkup(
    ToolName => 'Distributions',
    DFMDefinitionData => dfm_definition_data,
    DFMfields => dfm_fields);
end;
/

```

Enable descriptive flex fields

Generate a new ACD configuration to enable descriptive flex fields.

Example:

```

declare
    dfm_definition_data mverp_dfm_util.DFMDefinitionData_R;
    dfm_fields mverp_dfm_generate.GenerateFields_T;
begin
    -- extract current DFM data
    mverp_dfm_generate.GetToolDfmDefinition(
        ToolName => 'Distributions',
        DFMDefinitionData => dfm_definition_data,
        dfmfields => dfm_fields);
    -- set new values
    for i in nvl(dfm_fields.first, 0)..nvl(dfm_fields.last, -1)
    loop
        if dfm_fields(i).FieldName = MVERP_DFM_Util.DFM_FIELD_DFF then
            dfm_fields(i).DisplayFieldYN := 'Y';
            dfm_fields(i).SummaryFieldYN := 'Y';
            dfm_fields(i).ModifyFieldYN := 'Y';
        end if;
    end loop;
    -- save DFM data
    mverp_dfm_generate.GenerateExpressMarkup(
        ToolName => 'Distributions',
        DFMDefinitionData => dfm_definition_data,
        dfmfields => dfm_fields);
end;
/

```

Enable project fields

If the project fields are enabled in Oracle EBS, you must enable these fields in MarkView to edit lines.

Generate a new ACD configuration to enable the project fields, such as project, task, expenditure type, expenditure org, and date of expenditure.

Example:

```

declare
  dfm_definition_data mverp_dfm_util.DFMDefinitionData_R;
  dfm_fields mverp_dfm_generate.GenerateFields_T;
begin
  -- extract current DFM data
  mverp_dfm_generate.GetToolDfmDefinition(
    ToolName => 'Distributions',
    DFMDefinitionData => dfm_definition_data,
    dfmfields => dfm_fields);
  -- set new values
  for i in nvl(dfm_fields.first, 0)..nvl(dfm_fields.last, -1)
  loop
    if dfm_fields(i).FieldName in (
      MVERP_DFM_Util.DFM_FIELD_PROJECT,
      MVERP_DFM_Util.DFM_FIELD_TASK,
      MVERP_DFM_Util.DFM_FIELD_EXP_TYPE,
      MVERP_DFM_Util.DFM_FIELD_EXP_ORG,
      MVERP_DFM_Util.DFM_FIELD_EXP_ITEM_DATE
    ) then
      dfm_fields(i).DisplayFieldYN := 'Y';
      dfm_fields(i).SummaryFieldYN := 'Y';
    end if;
  end loop;
  -- save DFM data
  mverp_dfm_generate.GenerateExpressMarkup(
    ToolName => 'Distributions',
    DFMDefinitionData => dfm_definition_data,
    dfmfields => dfm_fields);
end;
/

```

Enable Income Tax

To enable the Income Tax Region field, you must also enable the Income Tax Type field.

To enable editing for the Income Tax Region and Income Tax Type fields in MarkView Viewer Accounting and to display these fields in Additional Details, generate a new ACD configuration.

Example:

```

declare
  dfm_definition_data mverp_dfm_util.DFMDefinitionData_R;
  dfm_fields mverp_dfm_generate.GenerateFields_T;
begin
  -- extract current DFM data
  mverp_dfm_generate.GetToolDfmDefinition(
    ToolName => 'Distributions',
    DFMDefinitionData => dfm_definition_data,
    dfmfields => dfm_fields);
  -- set new values
  for i in nvl(dfm_fields.first, 0)..nvl(dfm_fields.last, -1)
  loop
    if dfm_fields(i).FieldName in (
      MVERP_DFM_Util.DFM_FIELD_TAX_REGION,
      MVERP_DFM_Util.DFM_FIELD_TYPE_1099
    ) then
      dfm_fields(i).DisplayFieldYN := 'Y';
    end if;
  end loop;
end;
/

```

```

        dfm_fields(i).SummaryFieldYN := 'Y';
    end if;
end loop;
-- save DFM data
mverp_dfm_generate.GenerateExpressMarkup(
ToolName => 'Distributions',
DFMDefinitionData => dfm_definition_data,
dfmfields => dfm_fields);
end;
/

```

Add Undistributed Amount

To display Undistributed Amount in the Accounting detail section and in Additional Details for invoice lines, generate a new ACD configuration with the ShowDistributionTotal parameter set to TRUE. By default, ShowDistributionTotal is TRUE.

To hide Undistributed Amount, generate a new ACD configuration with the ShowDistributionTotal parameter set to FALSE.

Example:

```

declare
    dfm_definition_data MVERP_DFM_Util.DFMDefinitionData_R;
    dfm_fields          MVERP_DFM_Generate.GenerateFields_T;
begin
    -- extract current DFM data
    MVERP_DFM_Generate.GetToolDfmDefinition(
        ToolName => 'Distributions',
        DFMDefinitionData => dfm_definition_data,
        dfmfields => dfm_fields);
    -- set new values
    dfm_definition_data.ShowDistributionTotal := TRUE;
    -- save DFM data
    MVERP_DFM_Generate.GenerateExpressMarkup(
        ToolName => 'Distributions',
        DFMDefinitionData => dfm_definition_data,
        DFMfields => dfm_fields);
end;
/

```

Use tax default hierarchy

To use the tax default hierarchy, you must enable tax classification codes.

If you enable the UseTaxDefaultHierarchy parameter, the default Tax Classification value is automatically set for the Invoice line in MarkView Viewer Accounting.

To use the default Tax Classification value, generate a new ACD configuration with the UseTaxDefaultHierarchy parameter set to TRUE. By default, UseTaxDefaultHierarchy is FALSE.

Example:

```

declare
    dfm_definition_data MVERP_DFM_Util.DFMDefinitionData_R;

```

```

dfm_fields          MVERP_DFM_Generate.GenerateFields_T;
begin
  -- extract current DFM data
  MVERP_DFM_Generate.GetToolDfmDefinition(
    ToolName => 'Distributions',
    DFMDefinitionData => dfm_definition_data,
    dfmfields => dfm_fields);
  -- set new values
  dfm_definition_data.UseIlliTaxMode := TRUE;
  dfm_definition_data.UseTaxDefaultHierarchy := TRUE;
  -- save DFM data
  MVERP_DFM_Generate.GenerateExpressMarkup(
    ToolName => 'Distributions',
    DFMDefinitionData => dfm_definition_data,
    DFMfields => dfm_fields);
end;
/

```

Allow post validation changes

To edit distribution lines that are already validated but not yet approved, generate a new ACD configuration with the `AllowPostValidationChanges` parameter set to `TRUE`. By default, `AllowPostValidationChanges` is `FALSE`.

Example:

```

declare
  dfm_definition_data MVERP_DFM_Util.DFMDefinitionData_R;
  dfm_fields          MVERP_DFM_Generate.GenerateFields_T;
begin
  -- extract current DFM data
  MVERP_DFM_Generate.GetToolDfmDefinition(
    ToolName => 'Distributions',
    DFMDefinitionData => dfm_definition_data,
    dfmfields => dfm_fields);
  -- set new values
  dfm_definition_data.AllowPostValidationChanges := TRUE;
  -- save DFM data
  MVERP_DFM_Generate.GenerateExpressMarkup(
    ToolName => 'Distributions',
    DFMDefinitionData => dfm_definition_data,
    DFMfields => dfm_fields);
end;
/

```

Enable posted distribution lines changes

Note Before you configure this parameter, you must set `AllowPostValidationChanges` to `TRUE`.

To edit distribution lines that are already validated and posted in the general ledger, generate a new ACD configuration with the `EnablePostedLineChanges` parameter set to `TRUE`. By default, `EnablePostedLineChanges` is `FALSE`.

Example:

```

declare
  dfm_definition_data MVERP_DFM_Util.DFMDefinitionData_R;
  dfm_fields          MVERP_DFM_Generate.GenerateFields_T;
begin
  -- extract current DFM data
  MVERP_DFM_Generate.GetToolDfmDefinition(
    ToolName => 'Distributions',
    DFMDefinitionData => dfm_definition_data,
    dfmfields => dfm_fields);
  -- set new values
  dfm_definition_data.AllowPostValidationChanges := TRUE;
  dfm_definition_data.EnablePostedLineChanges := TRUE;
  -- save DFM data
  MVERP_DFM_Generate.GenerateExpressMarkup(
    ToolName => 'Distributions',
    DFMDefinitionData => dfm_definition_data,
    DFMfields => dfm_fields);
end;
/

```

Use withholding tax default hierarchy

Note In MarkView Viewer, you cannot add or edit withholding taxes for invoice lines and distributions.

If UseWHTDefaultHierarchy is TRUE, the withholding tax and the payment withholding tax values are automatically populated as specified for the invoice header. The distribution line withholding tax values are automatically populated as specified for the invoice line.

To use the withholding default tax hierarchy, generate a new ACD configuration with the UseWHTDefaultHierarchy parameter set to TRUE. By default, UseWHTDefaultHierarchy is FALSE.

Example:

```

declare
  dfm_definition_data MVERP_DFM_Util.DFMDefinitionData_R;
  dfm_fields          MVERP_DFM_Generate.GenerateFields_T;
begin
  -- extract current DFM data
  MVERP_DFM_Generate.GetToolDfmDefinition(
    ToolName => 'Distributions',
    DFMDefinitionData => dfm_definition_data,
    dfmfields => dfm_fields);
  -- set new values
  dfm_definition_data.UseWHTDefaultHierarchy := TRUE;
  -- save DFM data
  MVERP_DFM_Generate.GenerateExpressMarkup(
    ToolName => 'Distributions',
    DFMDefinitionData => dfm_definition_data,
    DFMfields => dfm_fields);
end;
/

```

Custom account validation

If you need to disable the use of the custom account validation, generate a new ACD configuration with the `UseCustomAccountValidation` parameter set to `FALSE`. By default, `UseCustomAccountValidation` is `TRUE`.

Example:

```
declare
  dfm_definition_data MVERP_DFM_Util.DFMDefinitionData_R;
  dfm_fields          MVERP_DFM_Generate.GenerateFields_T;
begin
  -- extract current DFM data
  MVERP_DFM_Generate.GetToolDfmDefinition(
    ToolName => 'Distributions',
    DFMDefinitionData => dfm_definition_data,
    dfmfields => dfm_fields);
  -- set new values
  dfm_definition_data.UseCustomAccountValidation := TRUE;
  -- save DFM data
  MVERP_DFM_Generate.GenerateExpressMarkup(
    ToolName => 'Distributions',
    DFMDefinitionData => dfm_definition_data,
    DFMfields => dfm_fields);
end;
/
```

Chapter 6

Make other updates after applying an Oracle Applications patch

Redo modifications

When an Oracle Applications patch is applied to the system, it is often a standard implementation without modifications. Therefore, some integration changes may be lost if an Oracle Applications patch is applied. While the exact changes are patch-dependent, you should verify and, if necessary, complete the following tasks after applying any Oracle Applications patch:

Reapply modifications to SFXINWKB

To apply modifications to the latest version of the form:

1. Back up your existing **SFXINWKB.fmb** file (located in `$_c_MARKVIEW_TOP/forms/US`) to a safe location.
2. Copy the new **APXINWKB.fmb** file to `$_c_MARKVIEW_TOP/forms/US`.
3. Rename the **APXINWKB.fmb** file as **SFXINWKB.fmb**.
4. Integrate the new version of the form.

Reapply modifications to Expense

To reapply your modifications to the Expense, rerun the Oracle Objects installer and re-install the `AP_WEB_OA_CUSTOM_PKG`:

1. On the Oracle E-Business Suite server, navigate to `<distribution_directory>/installer/bin`, where `<distribution_directory>` is the temporary distribution directory.
2. Run `./install.sh` to start the installer.
3. Select **OA Framework Integration for Expense**.
4. Follow the prompts for a Partial Install.
5. Re-install the `AP_WEB_OA_CUSTOM_PKG` in the Oracle Applications schema.
6. Reconfigure Expense Management. See [Expense Management Configuration](#).

Observe best practices

After applying an Oracle patch update, we strongly recommend that you follow the same best practices prescribed in the "Software Support for Upgrades and Bug Fixes" section of your Software Support Terms and Conditions.

As part of your software support, you are entitled to technical support assistance for software upgrades and bug fixes to troubleshoot and diagnose any incidents which may arise. All incidents will be addressed through the software support process described in the Software Support Terms and Conditions.

Before upgrading, consider contacting Kofax to discuss options for analyzing the overall impact of these product upgrades on the MarkView implementation.

It is required policy that all system changes be applied in your test environment and be tested thoroughly before migrating such system changes into a production environment.

For more information, see the detailed list of steps and best practices outlined in your copy of Software Support Terms and Conditions. This is an overview of these procedures:

- Back up the TEST system.
- Perform the upgrade in the TEST environment.
- Unit test the fix.
- Perform a User Acceptance Test of the changed TEST system.
- Back up the PROD system.
- Install in PROD during scheduled system downtime.
- Test PROD system before normal Production uptime.

Appendix A

Third-party license agreement

BEA Public License Version 2.1

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

- "License" shall mean the terms and conditions of this agreement.
- "Licensor" shall mean BEA Systems, Inc.
- "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means
 - (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or
 - (ii) ownership of fifty percent (50%) or more of the outstanding shares, or
 - (iii) beneficial ownership of such entity.
- "You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License, including but not limited to each Contributor other than Licensor in such Contributor's role as a licensee for the Software.
- "Source Format" shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.
- "Object Format" shall mean any form resulting from mechanical transformation or translation of a Source Format, including but not limited to compiled object code, generated documentation, and conversions to other media types.
- "Software" shall mean the original version of the software accompanying this agreement as released by BEA, including in Source or Object Format, and also any documentation provided therewith.
- "Derivative Works" shall mean any work, whether in Source or Object Format, that is based on (or derived from) the Software and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Software and derivative works thereof.
- "Contribution" shall mean any work of authorship, including the original version of the Software and any modifications or additions to that Software or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Software by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Software, but excluding

communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

- "Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Software.
2. **Grant of Copyright License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Software and such Derivative Works in Source or Object Format. Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the foregoing copyright license.
 3. **Grant of Patent License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Software, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Software to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Software or a Contribution incorporated within the Software constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Software shall terminate as of the date such litigation is filed.
 4. **Redistribution.** You may reproduce and distribute copies of the Software or Derivative Works thereof in any medium, with or without modifications, and in Source or Object Format, provided that You meet the following conditions:
 - You must give any other recipients of the Software or Derivative Works a copy of this License; and
 - You must cause any modified files to carry prominent notices stating that You changed the files; and
 - You must retain, in the Source Format of any Derivative Works that You distribute, BEA's copyright notice, "© [Date] BEA Systems, Inc. All rights Reserved.", and all other copyright, patent, trademark, and attribution notices from the Source Format of the Software, excluding those notices that do not pertain to any part of the Derivative Works; and
 - You must affix to the Software or any Derivative Works in a prominent manner BEA's copyright notice, "© [Date] BEA Systems, Inc. All rights Reserved." whenever You distribute the Software or such Derivative Works in Object Format.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Software otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Software by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Software and reproducing the content of the NOTICE file.

- 7. Disclaimer of Warranty.** EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR REPRESENTATIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using and distributing the Software and assume all risks associated with Your exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations. Further, You understand that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that its Contribution does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to You for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, You hereby assume sole responsibility to secure any other intellectual property rights needed, if any.
- 8. Limitation of Liability.** EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NO CONTRIBUTOR SHALL HAVE ANY LIABILITY TO YOU FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE SOFTWARE OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
- 9. Accepting Warranty or Additional Liability.** Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Software, if You include the Software in a commercial product offering, You may do so only in a manner which does not create potential liability for any Contributor. Therefore, if You include the Software in a commercial product offering, You shall and hereby do agree to defend and indemnify each and every Contributor against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against such Contributor(s) to the extent caused by Your acts or omissions in connection with Your distribution of the Software in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify to receive indemnification from You, a Contributor must:

 - a.** promptly notify You in writing of such claim, and
 - b.** allow the You to control, and cooperate with the You in, the defense and any related settlement negotiations.

The Contributor indemnified by You may participate in any such claim at its own expense.