

Migrating a workflow to production

Once the workflow model, business event and business process have been thoroughly tested and the sponsors and functional users have signed off, it's time to move the workflow to production.

Preparing for the migration

1. Ensure that end users who will be involved in the process or who have roles in the workflow are properly trained.
2. Ensure the environment is in good working order and the workflow model's administrator is prepared to deal with end user calls.
3. Make any final changes in workflow model such as updating email addresses and activity roles (if sample addresses or roles were used for testing purposes).
4. Document the new workflow.
 - a. Create a directory for workflow documentation and a subdirectory for each workflow
 - b. Create the workflow documentation .pdf file for the model
 - c. Include code associated with the workflow – triggers, packages
 - d. Document the event definition from Banner
 - e. Note anything special to the workflow – users/roles/proxies added to make it work
 - f. Complete business process documentation
5. Change the model status from TEST to ACTIVE. Remember that changing the status will delete any workflow instances.
6. Give DBA list of tables that WFBANNER user will need access to for any new tables that would not have already been granted rights for this user.

Migration activities

Some of the migration activities can be accomplished using Workflow scripts and the Event Wizard. A few steps have to be completed manually.

Workflow scripts/ commands	Export, Extractwd, Import	Create roles, components, workflow models
(Unsupported) script	parameter_load_script.sql	Create Banner event
Event Wizard	Event Wizard	Create Workflow Business Event
Manual		Create Business Process, associate model with Business Event, add users, role associations and proxies

Migrating the Banner Event

In Banner test environment using SSH Secure Shell:

Logon as yourself into PL/SQL Developer and open a command window(sqlplus).

We have made our own version of parameter_load_script.sql called data extract.sql

1. Run parameter_load_script.sql (see Appendix B) to create Banner event set-up script. You will be asked to supply the following:
 - a. &&EVENT_NAME = Banner event name
 - b. &&GROUP_NAME = Banner parameter group name

Run against Test, the I:\RISW\PLSQL Developer\WRKFLW migration\data extract.sql

Output goes to your C drive.

Code listing for data extract.sql

/*

This script can be run against the Event Queue Definition data in one instance of BANNER and then used to import an event into another instance.

The output of this script should be stored with any other import data, such as bootstraps, pl/sql procedures, table triggers and other things that will make up your workflow and allow it to function in another DB instance.

5/3/2001 -

Use command window - sqlplus

*/

```
undefine &&EVENT_NAME
undefine &&GROUP_NAME
set serveroutput on
set head off
set pagesize 1000
set linesize 1000
set verify off
set echo off
set feedback off
spool C:\event_&&EVENT_NAME.sql
select '-----' from dual;
select '--Parm Group Name Validation--' from dual;
select '-----' from dual;
select 'insert into gtveqpg values('' ||
GTVEQPG_CODE || ', '' ||
GTVEQPG_DESC || ', '' ||
GTVEQPG_USER_ID || ', to_date('' ||
GTVEQPG_ACTIVITY_DATE || ''));'
FROM GTVEQPG
WHERE GTVEQPG_CODE = '&&GROUP_NAME';
select '-----' from dual;
select '--Parm Name Validation --' from dual;
select '-----' from dual;
select 'insert into gtveqpm values('' ||
GTVEQPM_CODE || ', '' ||
GTVEQPM_DESC || ', '' ||
GTVEQPM_USER_ID || ', to_date('' ||
```

SUNGARD® HIGHER EDUCATION

```
GTVEQPM_ACTIVITY_DATE| | ''));'  
FROM GTVEQPM, GOREQPG  
WHERE GOREQPG_EQPG_CODE = '&&GROUP_NAME'  
      AND GTVEQPM_CODE = GOREQPG_EQPM_CODE;  
select '-----' from dual;  
select '--Parm Group Definition Form--' from dual;  
select '-----' from dual;  
select 'insert into goreqpg values('' | |  
GOREQPG_EQPG_CODE| | ''', '' | |  
GOREQPG_EQPM_CODE| | ''', '' | |  
GOREQPG_SEQNO| | ''', '' | |  
GOREQPG_DEFAULT_VALUE| | ''', '' | |  
GOREQPG_TARGET_NAME| | ''', '' | |  
GOREQPG_USER_ID| | ''', to_date('' | |  
GOREQPG_ACTIVITY_DATE| | ''));'  
FROM GOREQPG  
WHERE GOREQPG_EQPG_CODE = '&&GROUP_NAME';  
select '-----' from dual;  
select '--Event Validation Table --' from dual;  
select '-----' from dual;  
select 'insert into gtveqnm values('' | |  
GTVEQNM_CODE| | ''', '' | |  
GTVEQNM_DESC| | ''', '' | |  
GTVEQNM_USER_ID| | ''', to_date('' | |  
GTVEQNM_ACTIVITY_DATE| | ''));'  
FROM GTVEQNM, GOREQNM  
WHERE GOREQNM_EQPG_CODE = '&&GROUP_NAME'  
      AND GTVEQNM_CODE = GOREQNM_EQNM_CODE;  
select '-----' from dual;  
select '--Event que name def form--' from dual;  
select '-----' from dual;  
select 'insert into goreqnm values('' | |  
GOREQNM_EQNM_CODE| | ''', '' | |  
GOREQNM_EQPG_CODE| | ''', '' | |  
GOREQNM_EQTS_CODE| | ''', '' | |  
'I' | | ''', '' | | --I SET THE STATUS TO 'I' SO IT ISN'T ACTIVE IN THE NEW  
SYSTEM  
GOREQNM_USER_ID| | ''', to_date('' | |  
GOREQNM_ACTIVITY_DATE| | ''));'  
FROM GOREQNM  
WHERE GOREQNM_EQPG_CODE = '&&GROUP_NAME';  
undefine EVENT_NAME  
undefine GROUP_NAME  
spool off  
*****
```

Produced: event_WF_FIN_PUR_FABMATCsql.lst which ran against Prod with unique constraint errors on already existing rows. We ignored these.

```
-----  
--Parm Group Name Validation--  
-----  
insert into gtveqpg values('PO', 'PO related  
information', 'LOCAL', to_date('23-OCT-07'));  
-----  
--Parm Name Validation --  
-----
```

SUNGARD® HIGHER EDUCATION

```
insert into gtveqpm values('DOCNUMBER','Document
Number','BASELINE',to_date('29-NOV-99'));
insert into gtveqpm values('EVENTNAME','Workflow Event Name
(required)','BASELINE',to_date('29-NOV-99'));
insert into gtveqpm values('PRODUCTTYPE','Workflow Product Type
(reqd)','BASELINE',to_date('29-NOV-99'));
insert into gtveqpm values('WORKFLOWSPECIFICNAME','Workflow Specific
Name (reqd)','BASELINE',to_date('29-NOV-99'));
-----
--Parm Group Definition Form--
-----
insert into goreqpg
values('PO','DOCNUMBER',4,'','','LOCAL',to_date('23-OCT-07'));
insert into goreqpg
values('PO','EVENTNAME',1,'','','LOCAL',to_date('23-OCT-07'));
insert into goreqpg values('PO','PRODUCTTYPE',2,'SCT
Banner','','LOCAL',to_date('23-OCT-07'));
insert into goreqpg
values('PO','WORKFLOWSPECIFICNAME',3,'','','LOCAL',to_date('23-OCT-
07'));
-----
--Event Validation Table --
-----
insert into gtveqnm values('WF_FIN_PUR_FABMATC','Recepting request
reminder','LOCAL',to_date('23-OCT-07'));
-----
--Event que name def form--
-----
insert into goreqnm
values('WF_FIN_PUR_FABMATC','PO','WORKFLOW','I','LOCAL',to_date('24-
OCT-07'));
```

This will create the event_*Banner_event_name*.sql file. **If you remember to commit.**

2. Copy the event sql file to Banner production.
3. Copy database trigger files to Banner production.

In Banner production environment:

1. Run Event Queue load procedure, event_*Banner_event_name*.sql. This will update the Banner Event tables and create the event in Banner.
2. Verify entries on Event Queue forms.
3. Apply and enable database triggers.
4. Move new procedures to production and compile.
5. Change Active indicator on GOREQNM when ready to start events and workflows. NOTE: Do not activate until all parts of new event and workflow have been moved to production.
6. **Schedule any jobs that need scheduled.**

This is actually the last step.

export

PATH=\$PATH:\$ORACLE_HOME/bin:\$ORACLE_HOME/opmn/bin:\$WORKFLOW_HOME/PROD:\$WORKFLOW_HOME/PROD/bin – or not, Darren may have fixed it.

Command we actually used:

./export wfroot password ../bootstraps/<filename>

The actual command we run is highlighted below change date before you run:

./export wfroot password ../bootstraps/fullexp20100415.xml

Output produced is ../bootstraps/fullexp20100415.zip

Wfroot user password is in the configuration.xml file in the <SecurityIntegration> section. It is the superuser but is neither an oracle nor unix id. Password is the same for the webservice user.

Note: some oracle user passwords are hard coded in this file

- f. **Before you unzip the file check to see if the staticdocuments directory exists under /u01/app/sghe/workflow80/bootstraps/ if it does do the rm command for each doc in that directory and then do the rmdir staticdocuments to remove the directory. If you do not do this then when you try the unzip you will get warning messages about absolute path spec....and want you to answer with an y(yes), n(no), A(all), N(none). Even though it unzips when you try to use it in the extractwd command it will not work properly and you have to re-extract the file from the fullexp...file.**
- g. **Unzip this file from bootstraps directory**(for UNIX users, the unzip utility is located in the /users/bin directory).

2. Extract the desired workflow model using extractwd command.
 - a. Create the extract file - The **extractwd** command creates an extract of each model individually including associated roles and components.

Example: extractwd source_file target_file model_name version_number

1. **Source file** (i.e. fullexp20060101.xml - output from the export
Note: if you created the export file as a zip file, you will need to extract the .xml file first)
2. **Target file** (new file name for individual model .xml file)
3. **Model name** (must be exactly as in workflow)
4. Workflow model **version** number
5. **–withoutDependencies**
 - without dependencies extracts only the workflow model. The default is with dependencies which also extracts roles and components for the model.

Example: extractwd fullexp20060101.xml myWorkflowModel.xml ModelName 0

Note: If your workflow model name includes spaces, use the `--shell` option which will prompt for each reply:

```
Example: extractwd -shell  
<Target> myWorkflowModel.xml  
<Source> fullexport20060101.xml  
<Workflow Definition> ModelName  
<Workflow Definition Version> 0  
<With Dependencies [Y]> Y
```

Example #2:

```
C:\workflow\wfprod\bin>extractwd -shell  
Target> welcomewf.xml  
Source> ..\bootstraps\fullexp20051218.xml  
Organization> SGHE  
Workflow Definition> WelcomeWF  
Workflow Definition Version> 0  
With Dependencies [Y]> Y
```

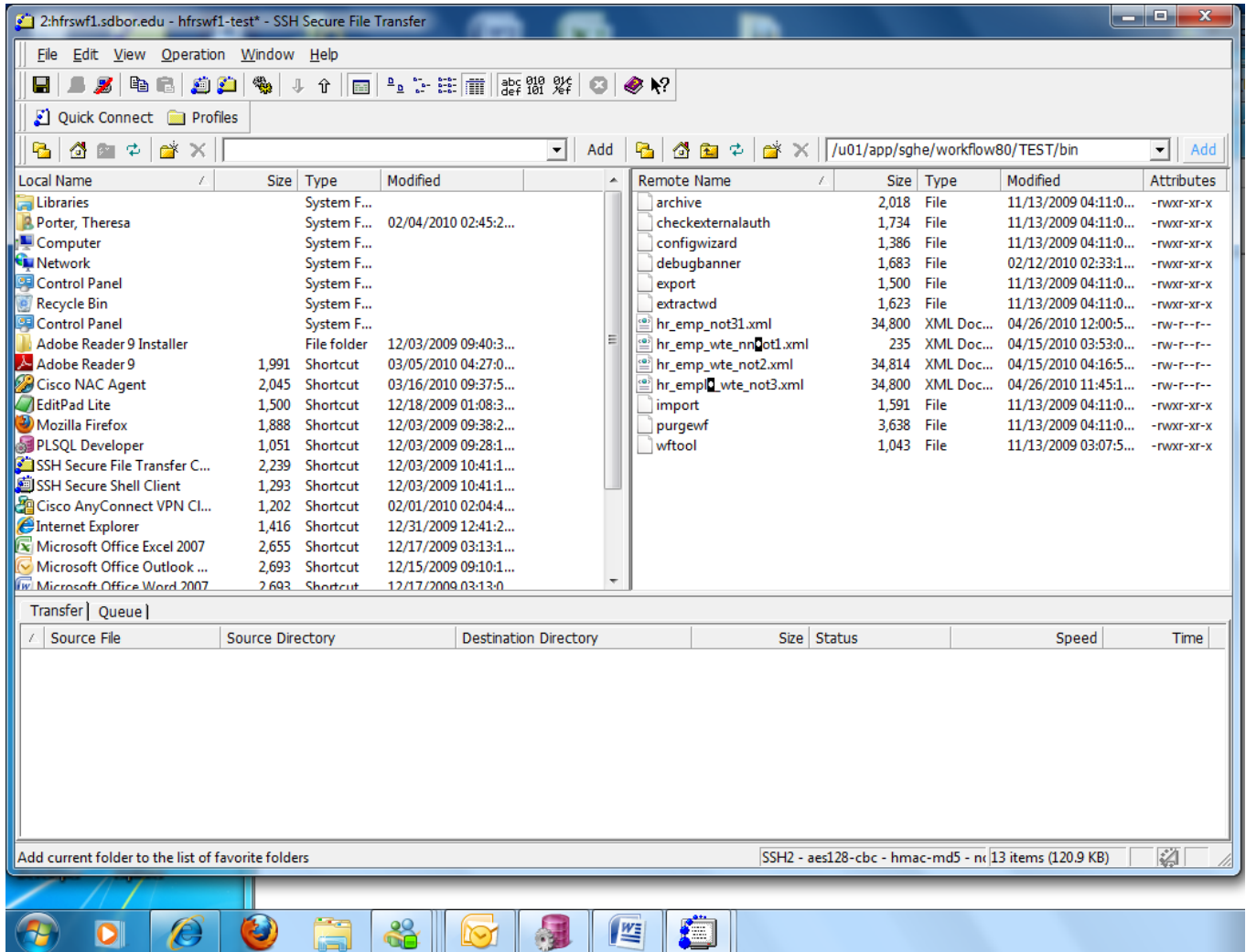
We used the `--shell` option successfully see below:

```
hfrswf1:/u01/app/sghe/workflow80/TEST/bin:=> extractwd -shell
```

```
Target> hr_emp_wte_not31.xml be careful typing backspaces etc here can cause  
special characters in the name see screenshot below.
```

```
Source> ../bootstraps/fullexp20100426.xml  
Organization> Root  
Workflow Definition> HR_Employee_WTE_Notification1  
Workflow Definition Version> 3  
With Dependencies [Y]> y --have to use lowercase y
```

If you run unsuccessfully then delete the Target file so you don't get two of them.



3. Copy workflow model xml file to Workflow production environment.
 - a. Copy file from TEST server to your hard drive using SSH Secure File Transfer (ftp) logged on as you.
 - b. Copy from your hard drive onto the PROD workflow server into your home directory /home/tmporter as an example using SSH Secure File Transfer(ftp) .
 - c. Then logon **as oracle user** from SSH Secure Shell and goto /u01/app/sghe/workflow80/PROD /bin then copy the file using the following command:
`cp /home/tmporter/hr_emp_wte_not31.xml .`

In Workflow production environment:

1. Import workflow model into production environment.
 - a. Go to the workflow_home directory for the correct environment (you should already be there from steps above)(i.e. prod) in our case PROD so path is /u01/app/sghe/workflow80/PROD
 - b. Go to the **.bin** directory of the target WF environment. `cd bin`

- c. Run the **import** using the output .xml file from the extractwd command

Example: import wfroot (password) myWorkflowModel.xml

import wfroot password hr_emp_wte_not31.xml

2. Review import error logs to ensure all objects you expected to move were imported correctly (roles, components). The log file will probably contain errors for objects that already existed in the new environment. Existing same-named objects are not overlaid.
 - a. Manually apply necessary updates to any existing objects that were not imported. For example, changes in components may need to be applied manually.
3. Change workflow model status to Active.

We were not able to find these log files
Used INB - GTVEQPM to check parameters

Updating Workflow Event and Business Process.

1. Use the Banner Event Wizard to build business event. See instructions above in this document.
 - a. Add workflow association to business event.
2. Build business process in Enterprise Management. –make sure Business Process status is set to active if this is a new workflow.
 - b. Add workflow association to business process. –if an existing workflow and you are promoting a new version put an end date on prior version and a begin date on the new version they cannot overlap.
 - c. Add event association to business process.

Other Setup

If necessary, add new workflow users, add roles assignments to users, and add proxy assignments. Notify appropriate offices of new workflow and required responses when they receive workflow activities.

Check groups, assign users.

Maintenance

Ensure that Workflow Engine is running. Periodically review Workflow logs to ensure workflow is starting and completing satisfactorily.

Turn on in INB using goreqnm (target system WORKFLOW)

Used INB – GOAEQRM to check events as well as workflow event status

Appendix B

Script to port Banner event definition data

parameter_load_script.sql

This script will extract the event data from Banner Event tables, and create a sql script to load the data into another instance. Run the following script in the original Banner instance, and run the created script in the new Banner instance.

<COPY FROM NEXT LINE>

/*

This script can be run against the Event Queue Definition data in one instance of BANNER and then used to import an event into another instance.

The output of this script should be stored with any other import data, such as bootstraps, pl/sql procedures, table triggers and other things that will make up your workflow and allow it to function in another DB instance.

5/3/2001 -

*/

```
undefine &&EVENT_NAME
undefine &&GROUP_NAME
set serveroutput on
set head off
set pagesize 1000
set linesize 1000
set verify off
set echo off
set feedback off
spool C:\event_&&EVENT_NAME.sql
select '-----' from dual;
select '--Parm Group Name Validation--' from dual;
select '-----' from dual;
select 'insert into gtveqpg values(''||
GTVEQPG_CODE||',''||
GTVEQPG_DESC||',''||
GTVEQPG_USER_ID||','to_date(''||
GTVEQPG_ACTIVITY_DATE||''))';
FROM GTVEQPG
WHERE GTVEQPG_CODE = '&&GROUP_NAME';
select '-----' from dual;
select '--Parm Name Validation --' from dual;
select '-----' from dual;
select 'insert into gtveqpm values(''||
GTVEQPM_CODE||',''||
```

SUNGARD® HIGHER EDUCATION

```
GTVEQPM_DESC||'',''|
GTVEQPM_USER_ID||'',''|to_date(''||
GTVEQPM_ACTIVITY_DATE||'');'
FROM GTVEQPM, GOREQPG
WHERE GOREQPG_EQPG_CODE = '&&GROUP_NAME'
  AND GTVEQPM_CODE = GOREQPG_EQPM_CODE;
select '-----' from dual;
select '--Parm Group Definition Form--' from dual;
select '-----' from dual;
select 'insert into goreqpg values(''||
GOREQPG_EQPG_CODE||'',''|
GOREQPG_EQPM_CODE||'',''|
GOREQPG_SEQNO||'',''|
GOREQPG_DEFAULT_VALUE||'',''|
GOREQPG_TARGET_NAME||'',''|
GOREQPG_USER_ID||'',''|to_date(''||
GOREQPG_ACTIVITY_DATE||'');'
FROM GOREQPG
WHERE GOREQPG_EQPG_CODE = '&&GROUP_NAME';
select '-----' from dual;
select '--Event Validation Table  --' from dual;
select '-----' from dual;
select 'insert into gtveqnm values(''||
GTVEQNM_CODE||'',''|
GTVEQNM_DESC||'',''|
GTVEQNM_USER_ID||'',''|to_date(''||
GTVEQNM_ACTIVITY_DATE||'');'
FROM GTVEQNM, GOREQNM
WHERE GOREQNM_EQPG_CODE = '&&GROUP_NAME'
  AND GTVEQNM_CODE = GOREQNM_EQNM_CODE;
select '-----' from dual;
select '--Event que name def form--' from dual;
select '-----' from dual;
select 'insert into goreqnm values(''||
GOREQNM_EQNM_CODE||'',''|
GOREQNM_EQPG_CODE||'',''|
GOREQNM_EQTS_CODE||'',''|
'I||'',''| --I SET THE STATUS TO 'I' SO IT ISN'T ACTIVE IN THE NEW SYSTEM
GOREQNM_USER_ID||'',''|to_date(''||
GOREQNM_ACTIVITY_DATE||'');'
FROM GOREQNM
WHERE GOREQNM_EQPG_CODE = '&&GROUP_NAME';
undefine EVENT_NAME
undefine GROUP_NAME
spool off
```