



COLLABORATE 18

TECHNOLOGY AND APPLICATIONS FORUM
FOR THE ORACLE COMMUNITY

What Version of Code Is Installed in This EBS Instance?

Session ID:

10198

Prepared by:

John Peters
Principal Consultant
JRPJR, Inc
john.peters@jrpjr.com

April 26, 2018, 8:30am, Breakers B

Remember to complete your evaluation for this session within the app!

#C18LV

About the Presenter

John Peters, JRPJR, Inc.

- Independent Consultant, working out of the SF Bay Area
 - Techno/Functional focus
 - Supply Chain Functional focus
 - Over 25 years working with Oracle EBS
 - Founding board member of the NorCal OAUG
 - OAUG APEX in EBS SIG Chairman
 - Member of the Oracle ATG CAB
- I enjoy sharing what I know about the Oracle E-Business Suite, I have many prior presentations posted on my web site at:
<http://jrpjr.com>

#C18LV

Abstract

- This presentation will show you tools and methods track your custom code and objects in an EBS environment. I will propose a process to allow you to keep track of: PL/SQL, Views, BI Publisher Elements in your EBS environments. Never question if a migration was completed again. I will provide scripts that allow you to compare multiple EBS environments simply to ensure everything has been migrated. I will also cover how Oracle Tracks their code versions and the EBS tools to track them.
- I am going to be focusing on code objects, not the tools themselves.
Example:
 - An actual RDF version, not the the version of the Oracle Reports tool

#C18LV

Agenda

- How does Oracle track their code versions
- How do I track code versions
- If you were to use ADOP things I would change or keep the same
- Summary of the reasons I am doing it differently

#C18LV

Oracle and Code Versions



COLLABORATE 18

TECHNOLOGY AND APPLICATIONS FORUM
FOR THE ORACLE COMMUNITY

Oracle's Code Versions

- Oracle tracks their code typically with a string 'Header' somewhere in a comment in the code itself.
- This applies to just about every object in the EBS.

#C18LV

How to Check Oracle's Code Versions

- How to check a files version:
 - `$AD_TOP/bin/adident Header <filename>`
 - Executables will show all of the included library versions (right hand screen shot)
 - Same as: `strings -a <filename> | grep Header`

#C18LV

```
appldev2@ebs12app-dev2:/oracle/OR12DEV2/apps/fs2/EBSapps/appl/inv/12.0.0/patch/115/sql
appldev2 |run| $ adident Header invmmttt.sql
invmmttt.sql:
$Header invmmttt.sql 120.9.12020000.2 2013/12/26 06:19:42 jianpyu ship $
appldev2 |run| $
```

```
appldev2@ebs12app-dev2:/oracle/OR12DEV2/apps/fs2/EBSapps/appl/fnd/12.0.0/bin
appldev2 |run| $ adident Header FNDLIBR
FNDLIBR:
$Header fdffvs.lc 120.31.12020000.12 2016/08/19 18:44:02 tebarnes ship $
$Header fdffvs.lc 120.31.12020000.12 2016/08/19 18:44:02 tebarnes ship $
$Header FNDLIBR.oc 120.0 2005/05/07 17:00:18 appldev ship $
$Header alstub.oc 120.0 2005/05/11 06:19:14 appldev ship $
$Header afpbwv.lpc 120.2.12020000.4 2012/08/10 15:44:13 ckclark ship $
$Header afpdon.lc 120.2 2005/09/02 18:30:05 pferguso ship $
$Header afpgrq.lpc 120.5.12020000.3 2016/05/11 17:18:10 tkamiya ship $
$Header afpi.lc 120.6.12020000.10 2016/02/26 18:28:00 tkamiya ship $
$Header afplog.lc 120.1 2005/08/25 23:37:50 aweisber ship $
$Header afpmir.lc 120.1 2009/08/05 15:50:12 jtoruno ship $
$Header afpolg.lpc 120.2.12020000.5 2016/02/26 18:33:10 tkamiya ship $
$Header afpo.lc 120.19 2010/04/21 21:11:59 pdeluna ship $
$Header afppio.lc 120.0 2005/05/07 17:03:57 appldev ship $
$Header afppir.lc 120.1.12020000.3 2013/08/05 21:24:19 pferguso ship $
$Header afpprr.lpc 120.8.12020000.6 2015/02/27 20:02:12 tkamiya ship $
$Header afprul.lpc 120.0.12020000.2 2012/07/13 20:14:48 pferguso ship $
$Header afprun.lc 120.0 2005/05/07 17:05:12 appldev ship $
$Header afpsec.lc 120.5 2010/05/05 11:01:17 mburra ship $
$Header afpsmc.lc 120.8.12020000.11 2015/03/18 20:51:25 tkamiya ship $
$Header afpsmg.lpc 120.8.12020000.2 2012/07/13 20:15:56 pferguso ship $
$Header afpsm.lc 120.4.12020000.2 2012/07/13 20:15:13 pferguso ship $
$Header afpsrs.lpc 120.8.12020000.5 2015/01/26 22:43:45 jtoruno ship $
$Header afstr.lc 120.0 2005/05/07 16:56:34 appldev ship $
$Header afsys.lc 120.3.12020000.6 2015/10/12 19:41:52 pferguso ship $
$Header aftime.lc 120.0 2005/05/07 16:56:37 appldev ship $
$Header afuddl.lc 120.2 2011/09/26 09:29:38 srinnakk ship $
```

How Else to Check Oracle's Code Versions

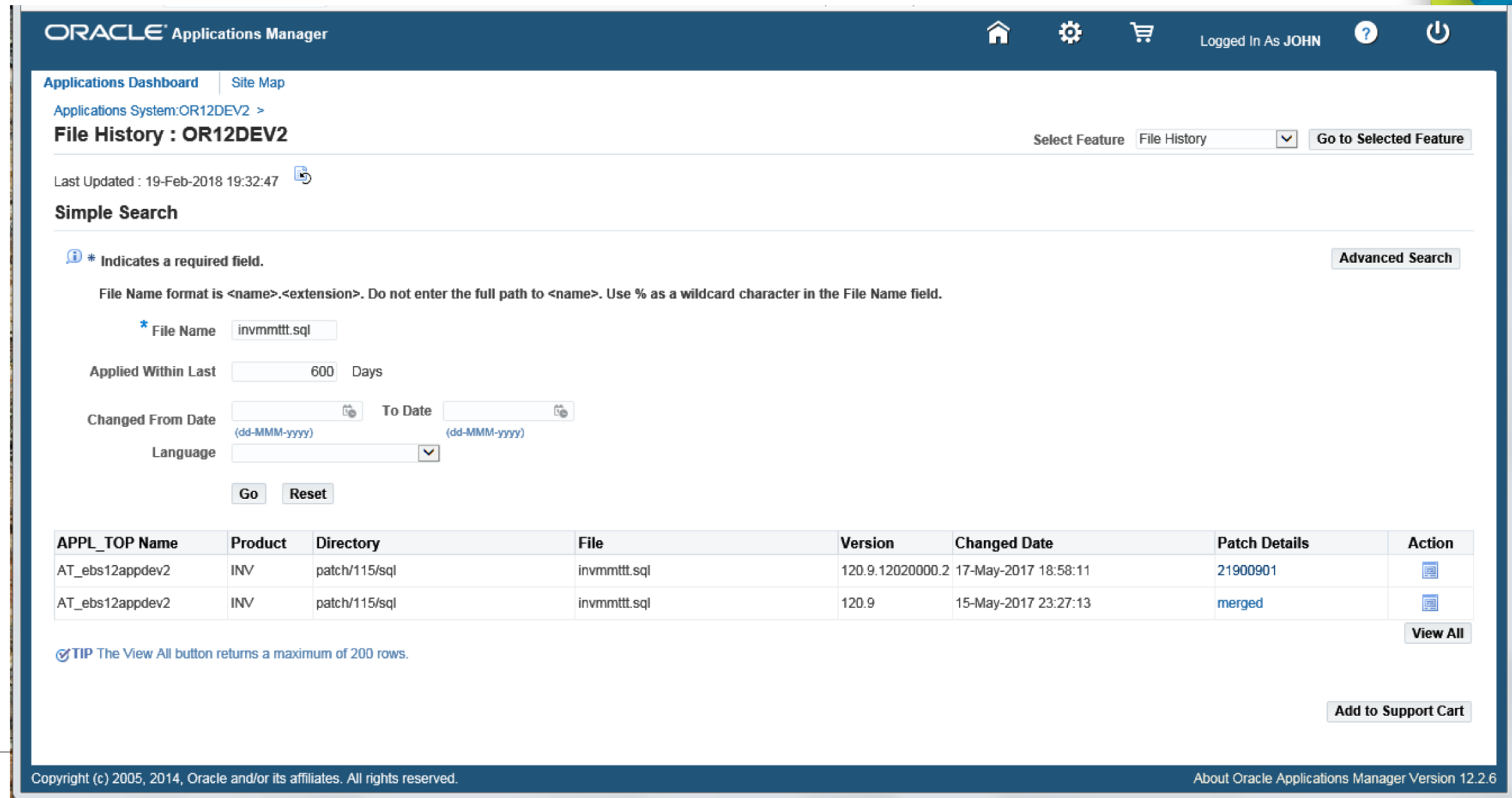
- Under System Administrator responsibility
- Oracle Applications Manager => Patching and Utilities

The screenshot displays the Oracle E-Business Suite interface. At the top, the header includes the Oracle logo, 'E-Business Suite', and user information 'Logged In As JOHN'. Below the header is an 'Enterprise Search' bar with a dropdown menu set to 'Contract Documents' and a 'Go' button. To the right of the search bar is a 'Search Results Display Preference' dropdown set to 'Standard'. The main content area is titled 'Home' and features a left-hand navigation menu under the 'System Administrator' folder. The menu items include 'Define Profile Options', 'Oracle Applications Manager' (which is expanded to show 'Workflow', 'License Manager', 'Service Fulfillment Manager', 'Patching and Utilities', 'Hosts', 'Purging/Critical Activities', 'System Alerts', 'Logs', and 'Applications Usage'), and 'Applications Usage'. The main content area on the right is titled 'Worklist' and contains a 'Full List' button, a toolbar with icons for refresh, back, forward, settings, and print, and a table with columns 'From', 'Type', 'Subject', 'Sent', and 'Due'. The table is currently empty, displaying the message 'There are no notifications in this view.' Below the table is a link for 'TIP Vacation Rules - Redirect or auto-respond to notifications.'

#C18LV

How Else to Check Oracle's Code Versions

- Select Feature => File History
- Enter File Name, Applied Within Last X days



The screenshot shows the Oracle Applications Manager interface for the 'File History' feature. The page title is 'File History : OR12DEV2'. The user is logged in as JOHN. The interface includes a search section with a 'Simple Search' form and an 'Advanced Search' button. The search form has fields for 'File Name' (containing 'invmttt.sql'), 'Applied Within Last' (600 Days), 'Changed From Date' and 'To Date' (both empty), and 'Language'. There are 'Go' and 'Reset' buttons. Below the search form is a table with columns: APPL_TOP Name, Product, Directory, File, Version, Changed Date, Patch Details, and Action. The table contains two rows of data. At the bottom, there is a 'View All' button and an 'Add to Support Cart' button. The footer contains copyright information and the version number 'About Oracle Applications Manager Version 12.2.6'.

ORACLE Applications Manager

Applications Dashboard | Site Map

Applications System:OR12DEV2 >

File History : OR12DEV2

Select Feature File History Go to Selected Feature

Last Updated : 19-Feb-2018 19:32:47

Simple Search

* Indicates a required field.

Advanced Search

File Name format is <name>.<extension>. Do not enter the full path to <name>. Use % as a wildcard character in the File Name field.

* File Name invmttt.sql

Applied Within Last 600 Days

Changed From Date To Date

Language

Go Reset

APPL_TOP Name	Product	Directory	File	Version	Changed Date	Patch Details	Action
AT_ebs12appdev2	INV	patch/115/sql	invmttt.sql	120.9.12020000.2	17-May-2017 18:58:11	21900901	
AT_ebs12appdev2	INV	patch/115/sql	invmttt.sql	120.9	15-May-2017 23:27:13	merged	

TIP The View All button returns a maximum of 200 rows.

Add to Support Cart

Copyright (c) 2005, 2014, Oracle and/or its affiliates. All rights reserved. About Oracle Applications Manager Version 12.2.6

How Else to Check Oracle's Code Versions

- Click on Action to view actions from that last file version

#C18LV

ORACLE Applications Manager Home Settings Shopping Cart Logged In As JOHN Help Logout

[Applications Dashboard](#) | [Site Map](#)

[Applications System:OR12DEV2](#) > [File History](#) >

Action Summary : u21900901.drv : OR12DEV2

Last Updated : 19-Feb-2018 19:34:21

Start Date	17-May-2017 14:39:36	End Date	17-May-2017 18:58:11
AutoPatch Options	hotpatch	Platform	LINUX
Driver File	u21900901.drv	Patch Top	/mnt/ipm_shared/stage/r1226_0916downloads/r1226/21900901

Filter is

i The details icon displays additional information about database updates.

Product	Directory	File	Action	Phase	Run	Bug Fix	Details
INV	patch/ 115/ sql	invmttt.sql	sql	en	Y	21900901	
INV	patch/ 115/ sql	invmttt.sql	copy		Y	21900901	

Copyright (c) 2005, 2014, Oracle and/or its affiliates. All rights reserved. About Oracle Applications Manager Version 12.2.6

How Else to Check Oracle's Code Versions

- Click on Details to view details for the actions
- These are all stored in tables for a given patches application

#C18LV

The screenshot displays the Oracle Applications Manager interface. At the top, the header includes the Oracle logo and 'Applications Manager', along with navigation icons and a user login status 'Logged In As JOHN'. Below the header, there are tabs for 'Applications Dashboard' and 'Site Map'. The main content area shows the breadcrumb 'Applications System:OR12DEV2 > File History > Action Summary >' followed by the title 'Action Details : invmmtt.sql : OR12DEV2'. Below this, it indicates 'Last Updated : 19-Feb-2018 19:35:35' and provides details for 'Bug Fixes 21900901', 'Run Y', and 'Action sql'. A metadata section lists 'Phase en', 'Product INV', and 'Directory patch/ 115/ sql'. The file name 'invmmtt.sql' is displayed. Below this, it shows 'Before Version N/A' and 'After Version N/A'. A table with 8 columns (Arguments, Command Modifier, Check Object, Elapsed Time, Start Time, Restart Time, End Time, Restarted?) contains one row of data. At the bottom right, there is an 'Add to Support Cart' button. The footer contains copyright information and the version 'About Oracle Applications Manager Version 12.2.6'.

ORACLE Applications Manager

Applications Dashboard | Site Map

Applications System:OR12DEV2 > File History > Action Summary >

Action Details : invmmtt.sql : OR12DEV2

Last Updated : 19-Feb-2018 19:35:35

Bug Fixes 21900901

Run Y

Action sql

Phase en

Product INV

Directory patch/ 115/ sql

invmmtt.sql

Before Version N/A

After Version N/A

Arguments	Command Modifier	Check Object	Elapsed Time	Start Time	Restart Time	End Time	Restarted?
N/A	sqlplus	none none none	0 sec	17-May-2017 18:02:44		17-May-2017 18:02:44	N

Add to Support Cart

Copyright (c) 2005, 2014, Oracle and/or its affiliates. All rights reserved. About Oracle Applications Manager Version 12.2.6

Why do they do it this way?

- I don't work for Oracle Development, but I would suspect that this all revolves around the fact that OS level files are how Oracle delivers patches and changes to code.
- The patching process needs a way to read an OS file, determine if that file is higher or lower than the one the patch is delivering and act accordingly.
- You must always be in the applmgr account at the OS Level to apply a patch.
- This is exactly why I needed an alternative method. Now with hosted environments and security concerns, rarely do we get applmgr or privileged DB accounts.

#C18LV

Another Method to Track Code Versions



COLLABORATE 18

TECHNOLOGY AND APPLICATIONS FORUM
FOR THE ORACLE COMMUNITY

Why track code versions differently?

- We don't use Oracle's Patching Tools to migrate custom code changes
 - Not using ADOP to migrate custom objects
 - Custom object patching is done in a down time window
- We don't have access to systems the way a DBA might be able to.
 - We don't have applmgr at OS level
 - We don't have a privileged DB account
 - We often don't even get OAM access
- There are times you might want get code versions from the code itself
Example:
 - if running version 2
then do B
else do A
- This is especially true with PL/SQL packages that interface to external systems

#C18LV

Custom PL/SQL Code



COLLABORATE 18

TECHNOLOGY AND APPLICATIONS FORUM
FOR THE ORACLE COMMUNITY

PL/SQL Code

- I put all code in a PL/SQL Package
 - No discrete standalone procedures and functions
- I put in the normal header revision comment block
- I also add a Function called PACKAGE_VERSION

```
1 • create or replace package body XXPRT_AME_UTILITY is
2   -- 2017/10/29  JPeters fixed the GET_PO_DIST_SOURCE function
3   -- 2018/02/05  JPeters added the GET_AME_PO_CMGR_APPROVER function
4   -- 2018/02/07  JPeters added routine to get XXPRT_PO_DEF_COMMODITY_MGR
5   -- 2018/02/19  JPeters added routine to get GET_PR_INV_ONLY
6
7
8   -----
9   FUNCTION PACKAGE_VERSION
10      RETURN VARCHAR2
11  IS
12 • BEGIN
13 •     RETURN '2018/02/19';
14 • END PACKAGE_VERSION;
15
16
```

How to Set PACKAGE_VERSION PL/SQL Function

#C18LV

- PACKAGE_VERSION just returns a string
- I like formatting it in YYYY/MM/DD format
- What if you make more than one change in a day?
Add a single letter after it A,B,C,D,E, etc
- This is just a label to help you know exactly what is installed in the DB
 - You have to be disciplined to fill in the PACKAGE_VERSION
- I like to keep at the top of the package body, right under the header block
- You could possibly have a source code control system insert the value

How to use PACKAGE_VERSION

- Just run an SQL statement to check package versions:

```
select d.NAME,  
       to_char(SYSDATE, 'YYYY/MM/DD HH24:MI') now,  
       xxprt_ame_utility.PACKAGE_VERSION xxprt_ame_utility,  
       xxprt_apex_auth_security.PACKAGE_VERSION xxprt_apex_auth_security,  
       xxprt_apex_security_pkg.PACKAGE_VERSION xxprt_apex_security_pkg,  
       xxprt_apex_upload.PACKAGE_VERSION xxprt_apex_upload,  
       xxprt_ar_util_pkg.PACKAGE_VERSION xxprt_ar_util_pkg,  
       xxprt_cnv_abc.PACKAGE_VERSION xxprt_cnv_abc,  
       xxprt_cnv_ap_invoice.PACKAGE_VERSION xxprt_cnv_ap_invoice,  
       xxprt_cnv_asl.PACKAGE_VERSION xxprt_cnv_asl,  
       xxprt_cnv_inv_onhand.PACKAGE_VERSION xxprt_cnv_inv_onhand  
from v$database d;
```

Save the output to Excel

	A	B	C	D	E	F	G	H	I
1	NAME	NOW	XXPRT_AME_UTILITY	XXPRT_APEX_SECURITY_PKG	XXPRT_APEX_UPLOAD	XXPRT_CNV_ABC	XXPRT_CNV_AP_INVOICE	XXPRT_CNV_ASL	XXPRT_CNV_INV_ONHAND
2	DEV	2017/09/19 10:25	2017/08/17	2017/01/21	2017/06/30	2017/09/01B	2017/08/25	2017/08/08	2017/08/28
3	TST	2017/09/19 10:20	2017/08/17	2017/01/21	2017/06/30	2017/09/01	2017/08/25	2017/08/08	2017/08/28
4	PRJ	2017/09/19 10:13	2017/08/17	2017/01/21	2017/06/30	2017/08/15	2017/08/25	2017/08/08	2017/08/28
5	PRD	2017/09/19 10:45	2017/08/17	2017/01/21	2017/06/30	2017/08/30	2017/08/25	2017/08/08	2017/08/28

How To Generate That SQL?

- The simple version of the SQL is:

```
select lower(OBJECT_NAME) || '.' ||  
       'PACKAGE_VERSION ' ||  
       lower(OBJECT_NAME) || ','  
from DBA_OBJECTS  
where OBJECT_NAME like 'XXPRT%'  
       and OBJECT_TYPE = 'PACKAGE'  
order by OBJECT_NAME, OBJECT_TYPE;
```

- You can get more sophisticated looking for PACKAGE_VERSION as a function, etc.
- I would actually argue that you only do this once, then maintain the check package version script with every new migration.

#C18LV

Benefits

- You can get the version of the PL/SQL even if the PL/SQL code is obfuscated using the wrap functionality
- You can get the version of the PL/SQL from remote databases calling through a DB link

#C18LV

Custom Views



COLLABORATE 18

TECHNOLOGY AND APPLICATIONS FORUM
FOR THE ORACLE COMMUNITY

Views

- Views are not as simple as the PL/SQL Package
- A colleague of mine, Dave Corey, came up with a creative solution.
- Store the date string in a comment in the DDL before the first column definition.
- The Comment becomes part of the view definition stored in ALL_VIEWS
- Same Header in DDL as Package

```
td_package_view_version_checker.sql *  XRPRT_VF_WOS_V.SQL *  +
9      -- MODIFICATION HISTORY
10     -- =====
11     -- DATE          NAME          DESCRIPTION
12     -- -----
13     -- 2017/05/18 DCOREY          Initial Creation
14     -- 2017/07/29 DCOREY          Put Id in
15     -- 2017/11/14 DCOREY          Added column ORGANIZATION_ID
16     -- -----
17     --
18
19 • connect APPS/@@APPS_PASS@@@DB;
20
21 • create or replace view apps.XRPRT_VF_WOS_V as
22   select /* 2017/11/14 */
23         mp.ORGANIZATION_CODE,
24         mp.ORGANIZATION_ID,
25         we.WIP_ENTITY_NAME work_order_number,
26         we.WIP_ENTITY_ID,
27         we.DESCRPTION work_order_description,
```

#C18LV

How to Use View Version

- The following SQL extracts the view version data

```
select v.VIEW_NAME,  
       substr(v.TEXT_VC,  
             instr(v.TEXT_VC, '/*', 1) + 2,  
             instr(v.TEXT_VC, '*/', 1) - 2 - instr(TEXT_VC, '/*', 1)  
       ) view_version  
from ALL_VIEWS v  
where (v.VIEW_NAME like 'XXPRT%'  
       or v.VIEW_NAME like 'PO_HEADERS_XML'  
       )  
and instr(v.VIEW_NAME, '#') = 0  
order by v.VIEW_NAME;
```

- In Excel copy columns to rows add instance and date time (you could do this in SQL)

	A	B	C	D	E	F	G	H
1	NAME NOW	PO_HEADERS_XML	XXPRT_APEX_EMPLOYEES_V	XXPRT_APEX_EXCEL_FILES_V	XXPRT_APEX_MENUS_V	XXPRT_APEX_PO_APPR_WF_MAILER_V	XXPRT_APEX_RESP_INV_ORGS_V	
2	DEV	2017/09/26 19:15	2017/06/02	2017/11/11	2017/09/07	2017/08/29	2017/12/04	2017/01/21
3	TST	2017/09/26 19:20	2017/06/02	2017/10/25	2017/09/07	2017/08/29	2017/12/04	2017/01/21
4	PRJ	2017/09/26 19:24	2017/06/02	2017/10/05	2017/09/07	2017/08/29	2017/12/04	2017/01/21
5	PRD	2017/09/26 19:25	2017/06/02	2017/10/05	2017/09/07	2017/08/29	2017/12/04	2017/01/21

Benefits

- You can see what version of the view is installed with absolutely no ambiguity.
- With Oracle the views are loaded from ODF or XDF files.
 - ODF: Object Definition File
 - XDF: XML Definition File
- The Header string does not get loaded into the DB. You can't easily tell what is loaded without running the adodfcmp and FndXdfCmp utilities.

#C18LV

Custom BIP Templates

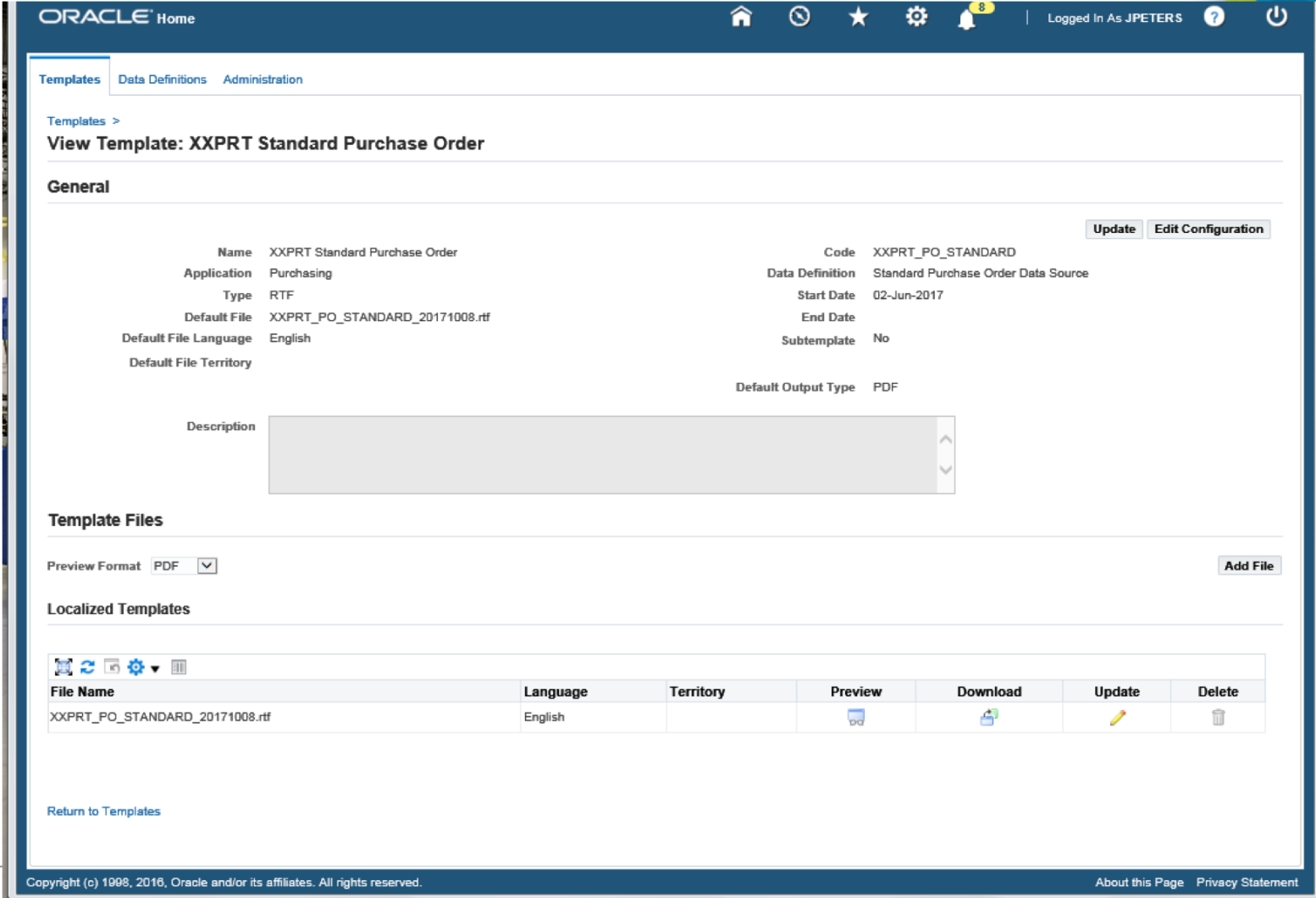


COLLABORATE 18

TECHNOLOGY AND APPLICATIONS FORUM
FOR THE ORACLE COMMUNITY

BIP Templates

- Put the data string on the end of the template filename
Example: `XXPRT_PO_STANDARD_20171008.rtf`
- Your developers have to use a standard convention
- This does not work for included or subtemplate files



The screenshot shows the Oracle BIP Templates administration interface. The page title is "View Template: XXPRT Standard Purchase Order". The "General" section contains the following details:

Name	XXPRT Standard Purchase Order	Code	XXPRT_PO_STANDARD
Application	Purchasing	Data Definition	Standard Purchase Order Data Source
Type	RTF	Start Date	02-Jun-2017
Default File	XXPRT_PO_STANDARD_20171008.rtf	End Date	
Default File Language	English	Subtemplate	No
Default File Territory		Default Output Type	PDF

The "Template Files" section shows a preview format of PDF and an "Add File" button. The "Localized Templates" section contains a table with the following data:

File Name	Language	Territory	Preview	Download	Update	Delete
XXPRT_PO_STANDARD_20171008.rtf	English					

At the bottom of the page, there is a "Return to Templates" link and a footer with the text "Copyright (c) 1998, 2016, Oracle and/or its affiliates. All rights reserved." and "About this Page Privacy Statement".

#C18LV

How to Use BIP Template Version

- You can extract this from the DB using the following SQL

```
select t.TEMPLATE_CODE,  
       l.FILE_NAME  
from XDO_LOBS l,  
     XDO_TEMPLATES_B t  
where l.LOB_TYPE in ('TEMPLATE', 'TEMPLATE_SOURCE')  
      and l.XDO_FILE_TYPE = 'RTF'  
      and t.TEMPLATE_CODE = l.LOB_CODE  
      and SYSDATE between t.START_DATE  
      and nvl(t.END_DATE, SYSDATE)  
      and t.TEMPLATE_CODE like 'XXPRT%'  
order by t.TEMPLATE_CODE;
```

- In Excel copy columns to rows add instance and date time (you could do this in SQL)

	A	B	C	D	E	F	G	H
1	NAME	NOW	XXPRT_AP_CHECK	XXPRT_BLANKET_REL	XXPRT_OEXOEAACK	XXPRT_PO_AGREEMENT	XXPRT_PO_STANDARD	XXPRT_PO_TC
2	DEV	2017/09/20 19:15	XXPRT_AP_CHECK_20170920.rtf	XXPRT_BLANKET_REL_20170919.rtf	SO_Ack_20170820.rtf	XXPRT_PO_AGREEMENT_20170919.rtf	XXPRT_PO_STANDARD_20170919.rtf	TC.rtf
3	TST	2017/09/20 19:25	XXPRT_AP_CHECK_20170701.rtf	XXPRT_BLANKET_REL_20170919.rtf	SO_Ack_20170820.rtf	XXPRT_PO_AGREEMENT_20170919.rtf	XXPRT_PO_STANDARD_20170919.rtf	TC.rtf
4	PRJ	2017/09/20 19:27	XXPRT_AP_CHECK_20170701.rtf	XXPRT_BLANKET_REL_20170919.rtf	SO_Ack_20170820.rtf	XXPRT_PO_AGREEMENT_20170919.rtf	XXPRT_PO_STANDARD_20170919.rtf	TC.rtf
5	PRD	2017/09/20 19:35	XXPRT_AP_CHECK_20170701.rtf	XXPRT_BLANKET_REL_20170919.rtf	SO_Ack_20170820.rtf	XXPRT_PO_AGREEMENT_20170919.rtf	XXPRT_PO_STANDARD_20170919.rtf	TC.rtf

#C18LV

Benefits

- You can see exactly what version of the BIP Template is installed with absolutely no ambiguity.
- You could easily expand this to include the row who for the BIP object.

#C18LV

Custom BIP Data Templates



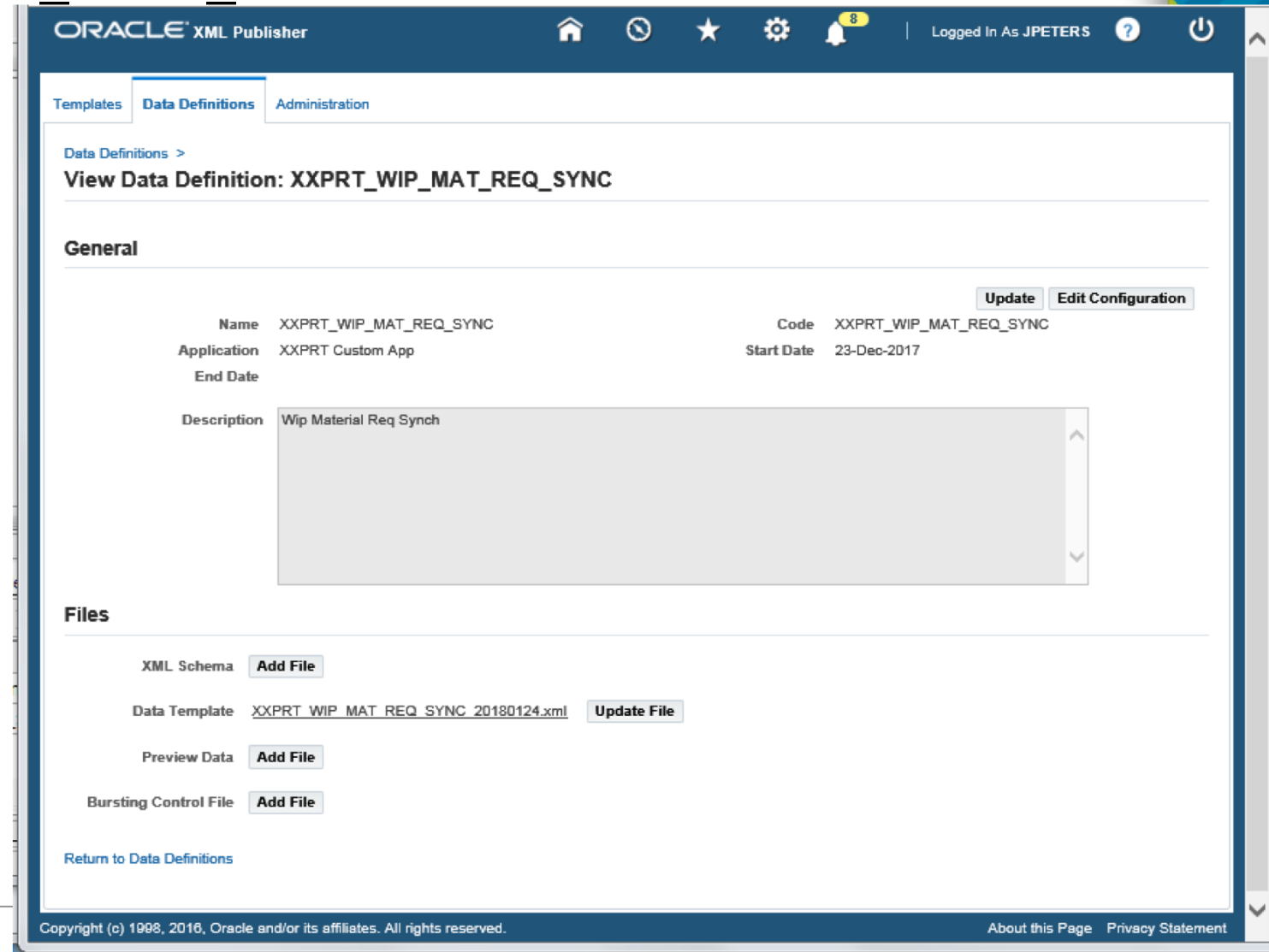
COLLABORATE 18

TECHNOLOGY AND APPLICATIONS FORUM
FOR THE ORACLE COMMUNITY

BIP Data Templates

- Put the data string on the end of the template filename
Example: `XXPRT_WIP_MAT_REQ_SYNC_20180124.xml`

- Your developers have to use a standard convention



The screenshot shows the Oracle XML Publisher interface. The top navigation bar includes 'ORACLE XML Publisher', a home icon, a clock icon, a star icon, a gear icon, a notification bell with '8', and a user profile 'Logged In As JPETERS'. The main content area has tabs for 'Templates', 'Data Definitions', and 'Administration'. The 'Data Definitions' tab is active, showing 'View Data Definition: XXPRT_WIP_MAT_REQ_SYNC'. Below this, there is a 'General' section with a table of metadata and a description field. The 'Files' section at the bottom contains several 'Add File' and 'Update File' buttons for different file types.

General	
Name	XXPRT_WIP_MAT_REQ_SYNC
Application	XXPRT Custom App
End Date	
Description	Wip Material Req Synch
Code	XXPRT_WIP_MAT_REQ_SYNC
Start Date	23-Dec-2017

Files

- XML Schema [Add File](#)
- Data Template `XXPRT_WIP_MAT_REQ_SYNC_20180124.xml` [Update File](#)
- Preview Data [Add File](#)
- Bursting Control File [Add File](#)

[Return to Data Definitions](#)

Copyright (c) 1998, 2016, Oracle and/or its affiliates. All rights reserved. [About this Page](#) [Privacy Statement](#)

#C18LV

How to Use BIP Data Template Version

- You can extract this from the DB using the following SQL

```
select s.DATA_SOURCE_CODE,  
       l.FILE_NAME  
from XDO_LOBS l,  
     XDO_DS_DEFINITIONS_B s  
where l.LOB_TYPE = 'DATA_TEMPLATE'  
      and s.DATA_SOURCE_CODE = l.LOB_CODE  
      and SYSDATE between s.START_DATE  
      and nvl(s.END_DATE, SYSDATE)  
      and s.DATA_SOURCE_CODE like 'XXPRT%'  
order by 1;
```

- In Excel copy columns to rows add instance and date time (you could do this in SQL)

	A	B	C	D
1	NAME	NOW	XXPRT_VF_PICK_SLIP	XXPRT_WIP_MAT_REQ_SYNC
2	DEV	2018/01/25 19:15	XXPRT_VF_PICK_SLIP_20170701.xml	XXPRT_WIP_MAT_REQ_SYNC_20180124.xml
3	TST	2018/01/25 19:17	XXPRT_VF_PICK_SLIP_20170701.xml	XXPRT_WIP_MAT_REQ_SYNC_20180124.xml
4	PRJ	2018/01/25 19:19	XXPRT_VF_PICK_SLIP_20170701.xml	XXPRT_WIP_MAT_REQ_SYNC_20180124.xml
5	PRD	2018/01/25 19:24	XXPRT_VF_PICK_SLIP_20170701.xml	XXPRT_WIP_MAT_REQ_SYNC_20180124.xml

#C18LV

Benefits

- You can see exactly what version of the BIP Data Template is installed with absolutely no ambiguity.
- You could easily expand this to include the row who for the BIP object.

#C18LV

Gaps

Things I Have Not Handled
(primarily because I don't need to)



COLLABORATE 18

TECHNOLOGY AND APPLICATIONS FORUM
FOR THE ORACLE COMMUNITY

Oracle Reports RDFs

- I am finding RDF development has really dropped off in recent years.
 - Talking no RDFs or at most 2
 - Never create a RDF from scratch
- If we modify an Oracle Seeded report I like to keep the Header string of the original file.
 - This allows us to know where the branched development came from
- Since these are OS level objects we don't typically have access to them anyway
 - End up on relying on the DBA
- When I have had to compare file versions I use the file size

Oracle Forms FRMs and OA Framework Pages

- It is rare when I have any Forms or OA Framework development done on my projects
- I use APEX almost exclusively for UI type development

#C18LV

Host Programs, OS Shell and PERL Scripts

- If I had to I would use a comment in the code (Header concept like Oracle does)
- Since these are OS level objects we don't typically have access to them anyway
 - End up on relying on the DBA
- When I have had to compare file versions I use the file size

#C18LV

Java or C Programs

- It is rare when I need to do Java development
- If I had to I would use a comment in the code (Header concept like Oracle does)
- Since these are OS level objects we don't typically have access to them anyway
 - End up on relying on the DBA

#C18LV

Using ADOP

What would I change



COLLABORATE 18

TECHNOLOGY AND APPLICATIONS FORUM
FOR THE ORACLE COMMUNITY

To Be Continued

- To Be Continued

#C18LV

Summary



COLLABORATE 18

TECHNOLOGY AND APPLICATIONS FORUM
FOR THE ORACLE COMMUNITY

**Please Complete Your
Session Evaluations**

Session ID: 10198



COLLABORATE 18

TECHNOLOGY AND APPLICATIONS FORUM
FOR THE ORACLE COMMUNITY



COLLABORATE 18

TECHNOLOGY AND APPLICATIONS FORUM
FOR THE ORACLE COMMUNITY

John.peters@jrpjr.com

<http://jrpjr.com>

Session ID:

10198

Remember to complete your evaluation for this session within the app!

#C18LV

Your future's bright...



**WHEN YOU RELY
ON THE OAUG TO
ILLUMINATE YOUR PATH!**

**VISIT BOOTH 1335 TO DISCOVER
HOW OAUG MEMBERS SHINE.**

OAUG™ OAUG.ORG/ILLUMINATE