



# 2021 **WINSHUTTLE**

Compare Extract Scripts  
For EnterWorks 10.x



# Compare Extract Scripts

For EnterWorks 10.x

## Table of Contents

Compare Extract Scripts.....	4
Procedure - Generate Full Report (MS SQL Server) .....	7
Procedure - Generate Full Report (PostgreSQL) .....	12
Procedure - Generate Partial EPX Report .....	18
Procedure – Comparing Extraction Reports Using WinMerge .....	22

## Compare Extract Scripts

The compare extract scripts will generate formatted reports from SQL Server Management Studio or pgAdmin4 (for PostgreSQL) of the various EnterWorks and EPX objects in a specific structured order to facilitate comparing two environments by comparing the corresponding extract files.

Each extract script has a recommend text field size (for MS SQL Server) to ensure the possible values are complete while not making the physical size (in terms of character width) excessively large. The following table lists the available scripts, their purpose, the recommended text size. The instructions for setting the text size are included below.

SQL Script	Purpose	Query Text Size (MS SQL Server)
CompareEnableServerExtract_CodeSets.sql (SQL Server) CompareEnableServerExtract_CodeSets.pg.sql (PostgreSQL)	Extracts the Code Sets, Taxonomy and Hierarchy definitions	8192
CompareEnableServerExtract_Core.sql (SQL Server) CompareEnableServerExtract_Core.pg.sql (PostgreSQL)	Extracts the most common objects, including groups, profiles, validation rules, repositories, and configuration repositories.	8192
CompareEnableServerExtract_ImportExport.sql	Extracts the import template, export template, syndication template, and publication template definitions	8192
CompareEnableServerExtract_SearchPref.sql	Extracts the User Preferences and defined advanced searches.	8192
CompareEnableServerExtract_Security.sql	Extracts the details for the attribute and repository security filters and the security settings for all groups and objects	8192
CompareEnableServerExtract_ServicesFramework.sql	Extracts only the model objects associated with the Services Framework	8192
CompareEnableServerExtract_SQL.sql	Extracts the tables, views and stored procedures from the EPIM database	8192
CompareEnableServerExtract_SQL_ServicesFramework.sql	Extracts the tables, views and stored procedures associated with the Services Framework from the EPIM database	8192
CompareEPXServerExtract.sql	Extracts the users, roles, and workflows from EPX	8192
CompareEPXServerExtract_ServicesFramework.sql	Extracts the workflows associated with the Services Framework	8192

When each script is executed for MS SQL Server and the results output to a file, it is recommended to follow a naming convention for those files such that the project and environment and specific content are identified using the following naming convention:

<type>\_<environment>\_<project>.rpt

Where:

- <type> - identifies the type of file: CodeSet, Core, ImportExport, SearchPref, Security, ServicesFramework, SQL, ServicesFrameworkSQL, EPX, ServicesFrameworkEPX
- <environment> - identifies the environment: DEV, QA, PROD
- <project> - identifies the project: CPO, ACPRO, Orgill, etc.

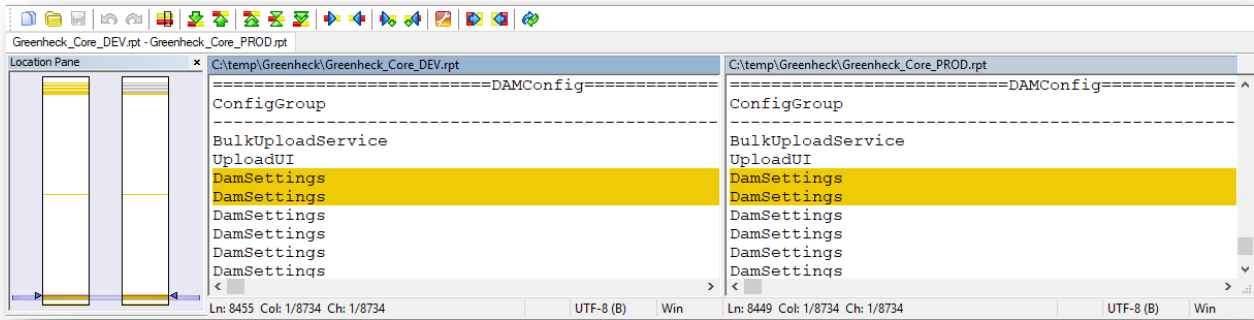
Once the files have been generated from each environment, they can be compared using a tool such as WinMerge or searched using a text editor.

Each file will generate sections that begin with a series of equal signs and identify the section, followed by more equal signs. For example, the Core report includes the following sections:

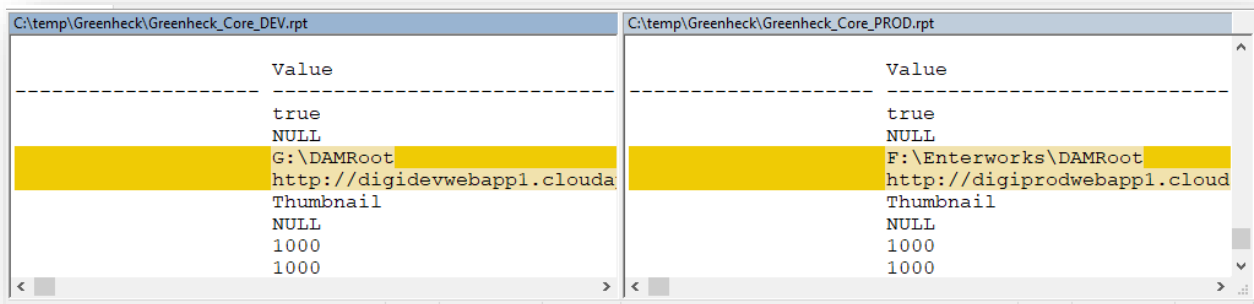
```
=====Groups=====
=====Group Capabilities=====
=====Group Home Page Widgets=====
=====Profiles=====
=====Validation Rules=====
=====Link Relationships=====
=====Repository Folders=====
=====Repositories=====
=====Repository Attributes=====
=====Code Sets=====
=====Scheduled Exports=====
=====Scheduled Imports=====
=====Package Promotions=====
=====Promotions=====
=====Shortcuts=====
=====CN_Registry=====
=====MQ_Registry=====
=====DAMConfig=====
=====DAMVariants=====
=====Automated Sort=====
=====Publication Merge=====
=====Sequences=====
=====Server Properties=====
=====Transmission Options=====
```

Each section has a set self-identifying fields. The rows in each section are ordered so differences are more easily identified by WinMerge.

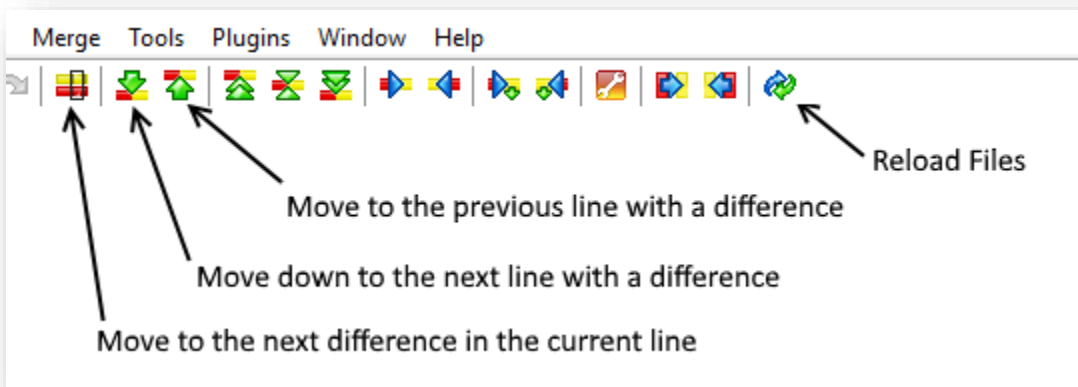
When two files for the same report (but different environments) are loaded into WinMerge, the differences between them are highlighted:



The Location Pane graphically shows where there are differences in the file. Clicking on a location will move the file view panes on the right to that location. Lines that have differences will be shaded. When the horizontal scrollbar is moved, the contents in both panes move together. The actual differences in the line will be shaded a different color:



The toolbar has several buttons that help navigate through the changes:



For details on the other icons, consult the WinMerge documentation or online help.

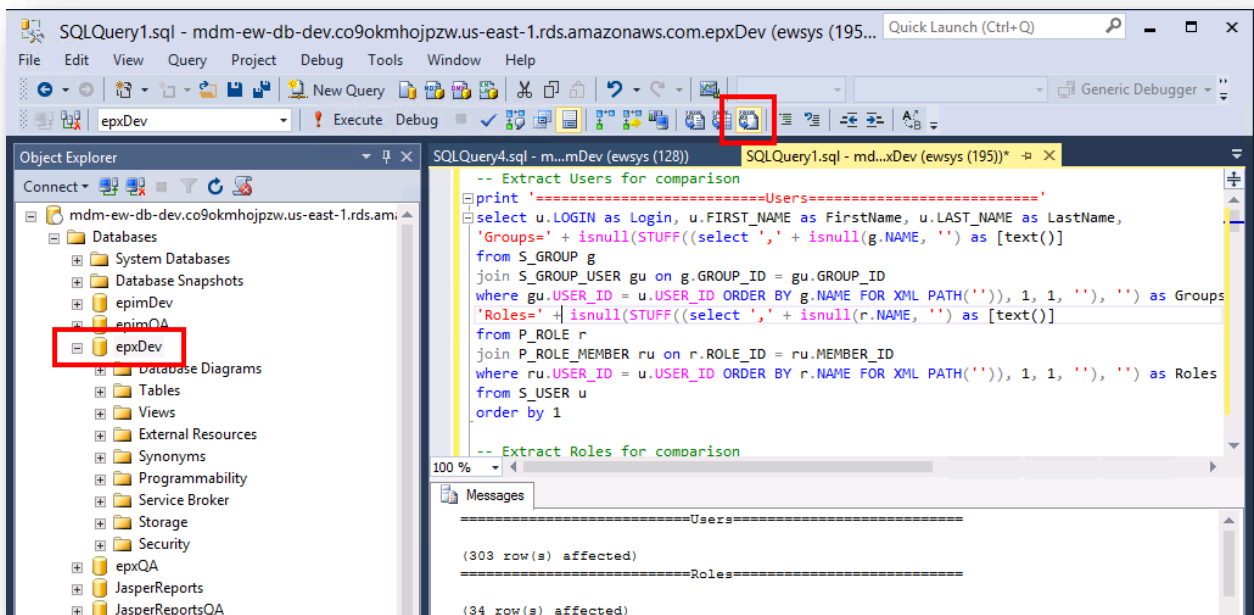
The following screen shot shows how the differences appear when using the navigation buttons to move up or down and to find the next difference in the current row:

C:\temp\Greenheck\Greenheck_Core_DEV.rpt	C:\temp\Greenheck\Greenheck_Core_PROD.rpt
Value	Value
true	true
NULL	NULL
G:\DAMRoot	F:\Enterworks\DAMRoot
http://digidevwebapp1.cloudapp.net/Damroot	http://digiprodwebapp1.cloudapp.net/Damroot
Thumbnail	Thumbnail
NULL	NULL
1000	1000
1000	1000

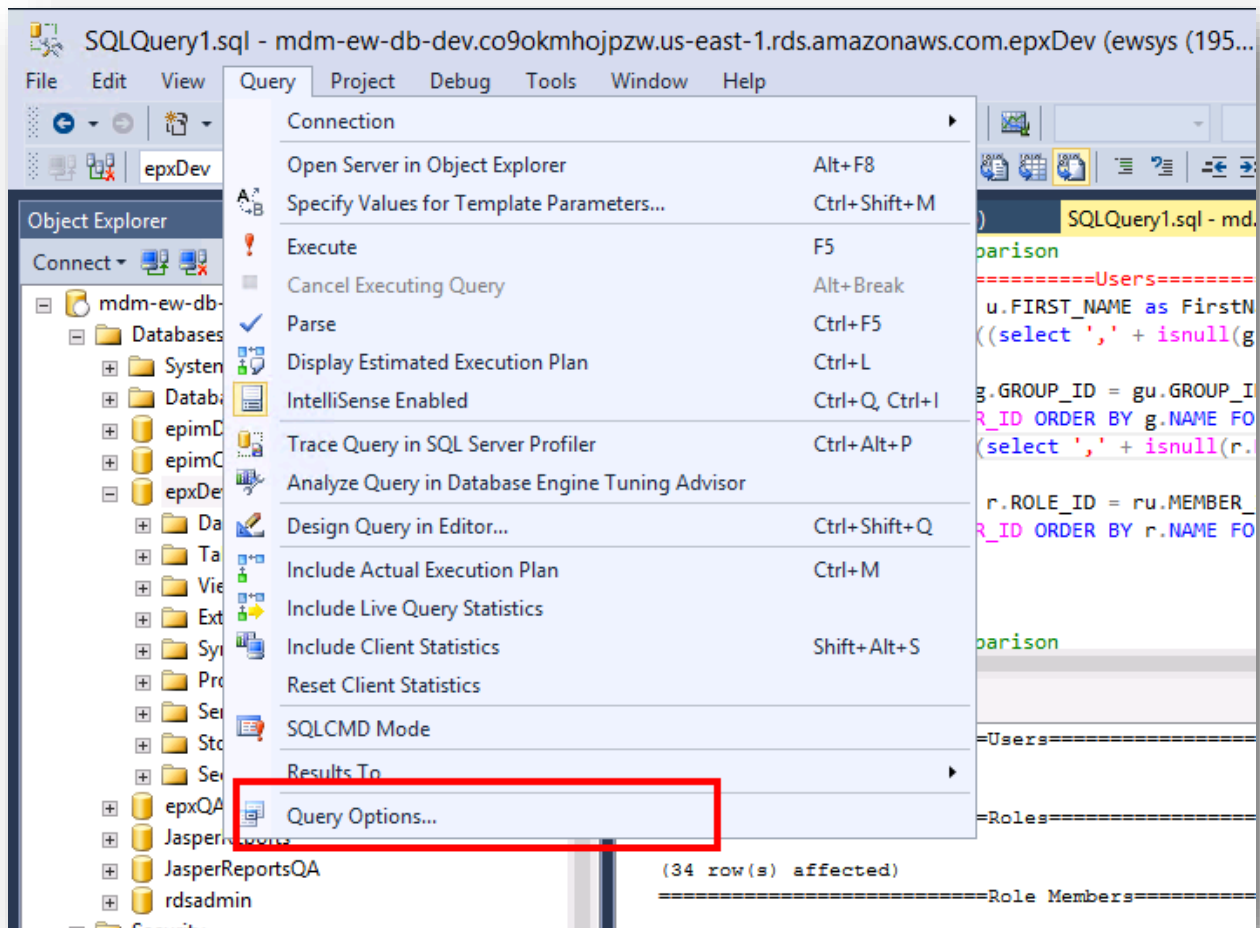
## Procedure - Generate Full Report (MS SQL Server)

To generate a report of all objects within an EPIM or EPX database in MS SQL Server for different environments (e.g., DEV vs. QA/PROD), perform the following steps:

1. Connect to the EPIM or EPX database using the SQL Server Management Studio application.
2. Open a New Query window on the EPIM or EPX database.
3. Load the desired compare extraction script into a new Query window.
4. Select the Results to File option:

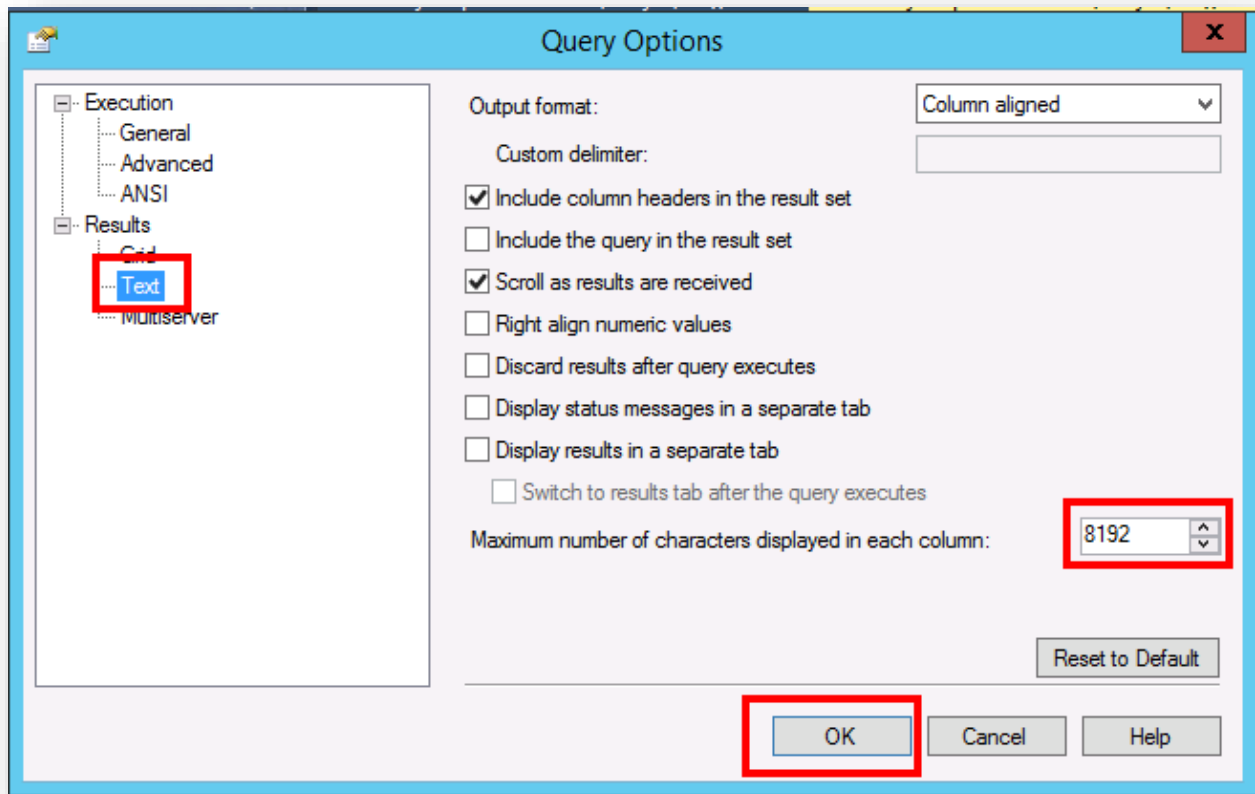


5. Select menu option Query->Query Options...:

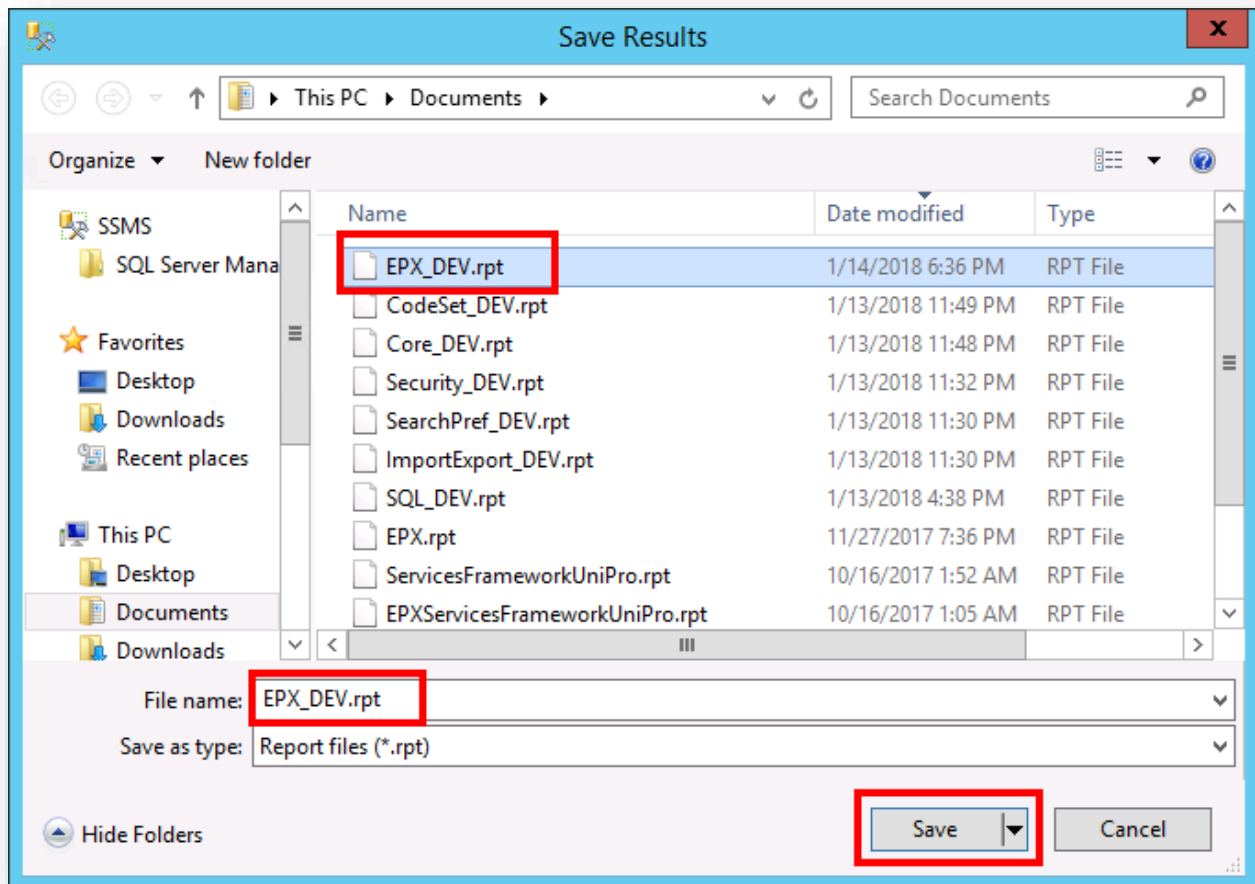


- Set the Maximum number of characters displayed in each column to 512 or 8192 (see table for the proper size for each compare extraction script) for Results->Text and click OK:

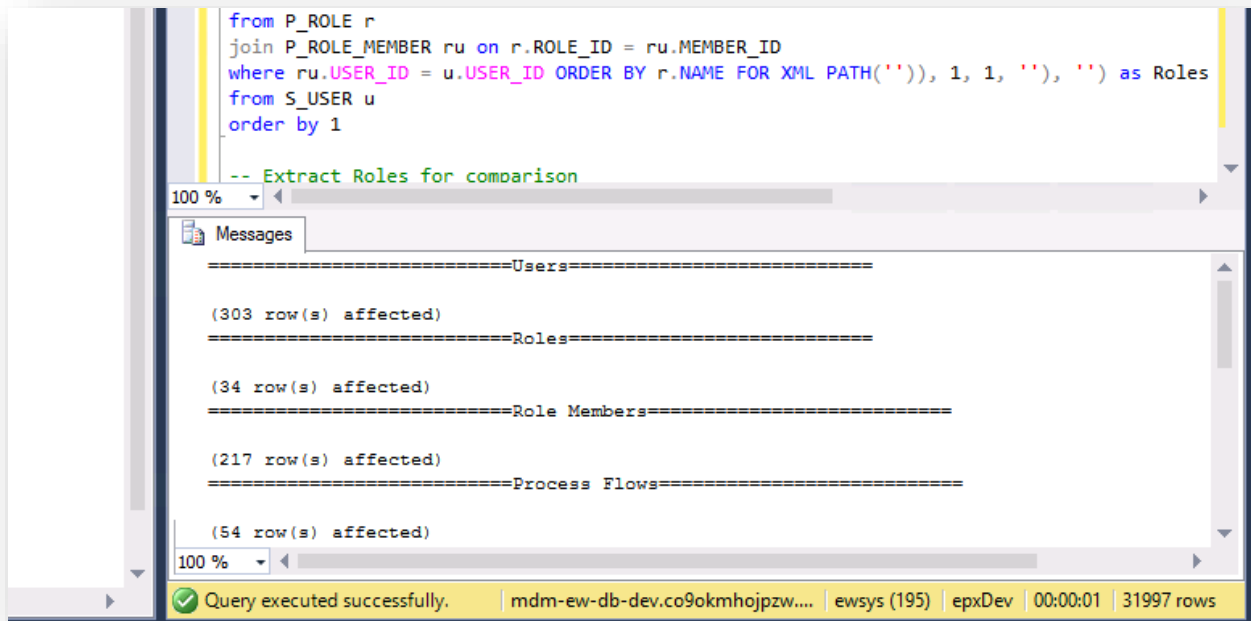




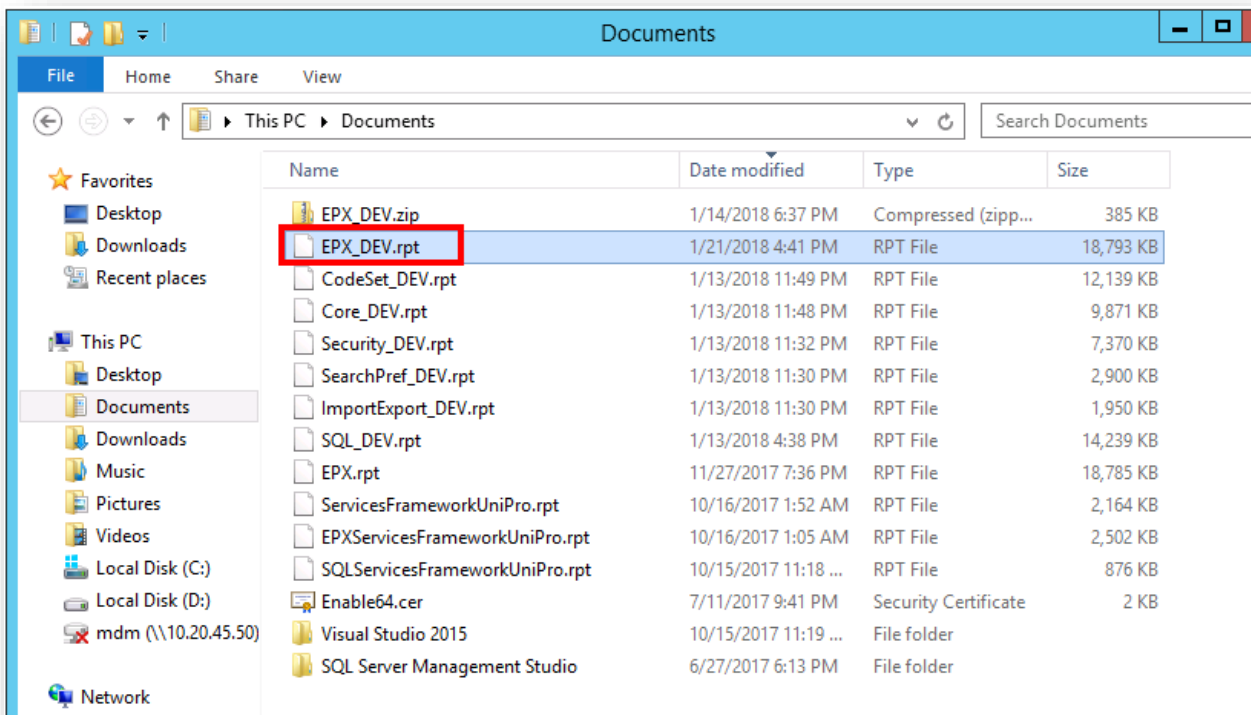
7. Execute the query. A prompt appears to select or specify the .rpt file to be created. Select or enter the name of the file and click Save:



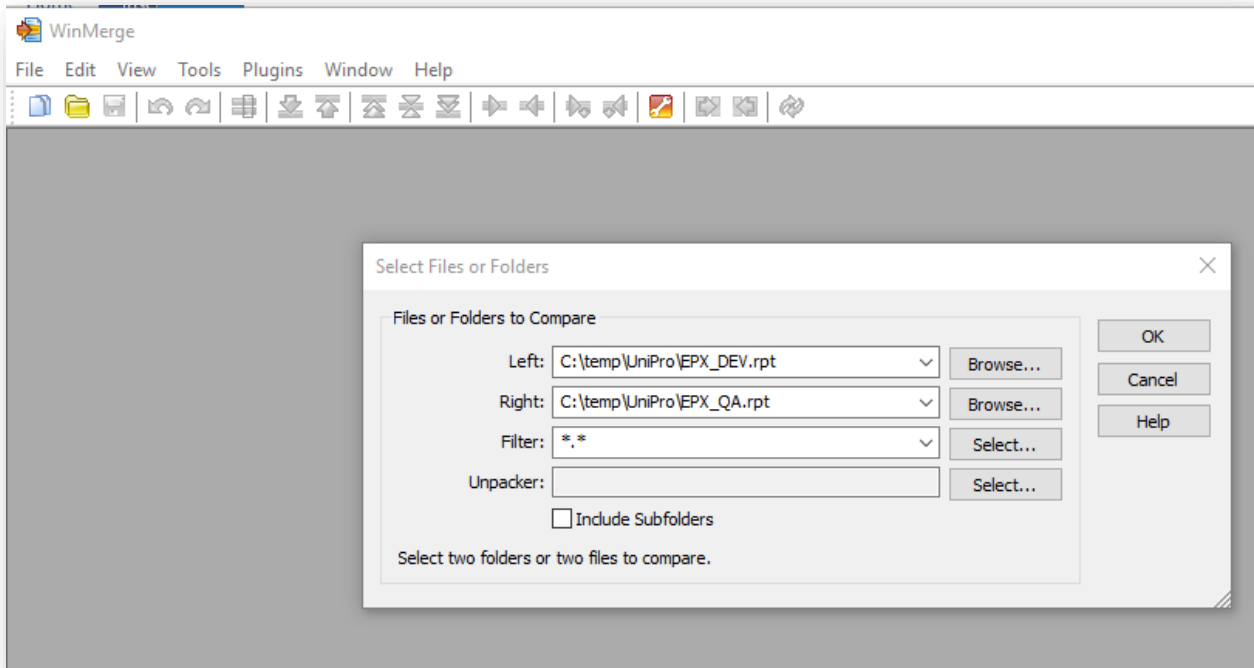
8. The Messages field will show the results of each query being executed:



- Copy the resulting .rpt file and use in comparisons and to update the source repository (e.g., Subversion or GIT) if versioning the data model contents:



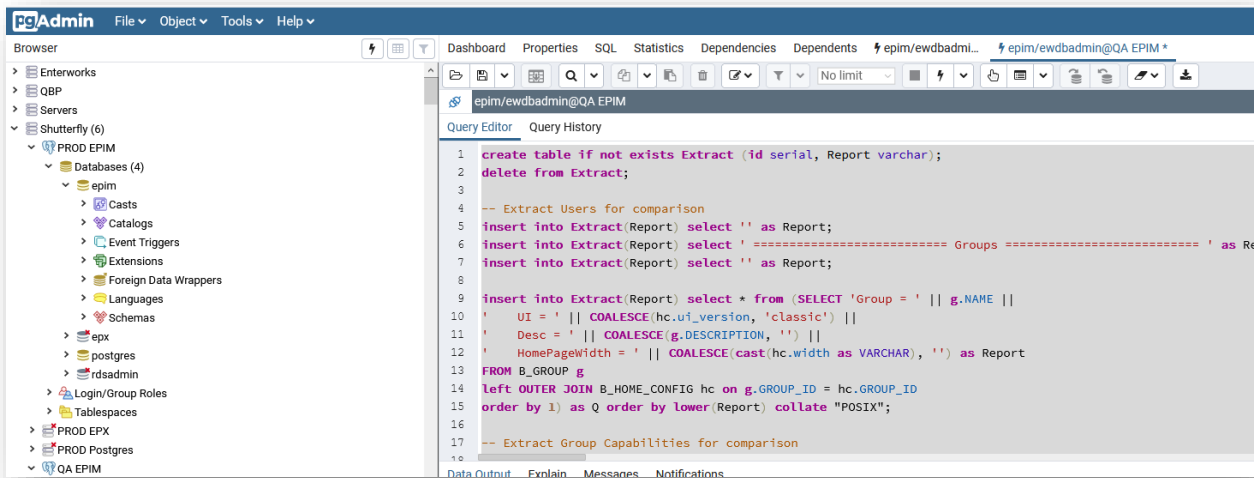
10. Repeat the above steps on the same database using the same script for the second environment.
11. Run the WinMerge utility to compare the two rpt files:



## Procedure - Generate Full Report (PostgreSQL)

To generate a report of all objects within an EPIM or EPX database in PostgreSQL for different environments (e.g., DEV vs. QA/PROD), perform the following steps:

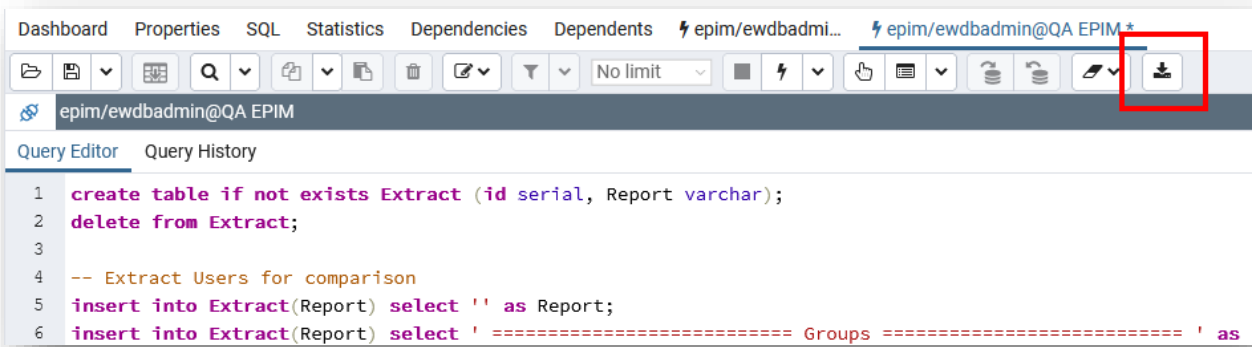
1. Connect to the EPIM or EPX database using the pgAdmin 4 application.
2. Open the Query tool window on the EPIM or EPX database.
3. Load the desired compare extraction script into a new Query Editor window:



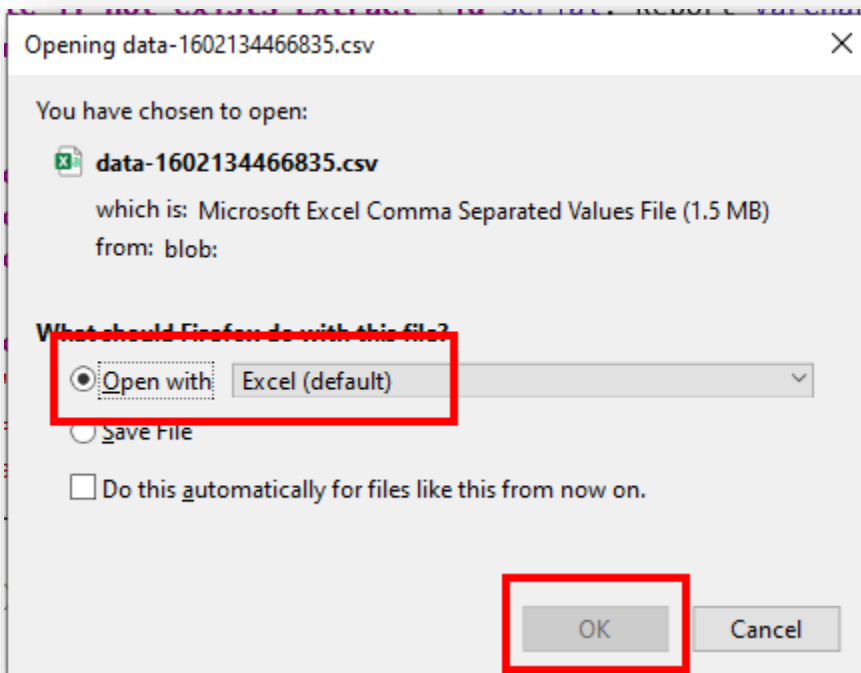
4. Execute the query. The results appear in the Query Editor window:

13	FROM B_GROUP g
14	left OUTER JOIN B_HOME_CONFIG hc on g.GROUP_ID = hc.GROUP_ID
15	order by 1) as Q order by lower(Report) collate "POSIX";
16	
17	-- Extract Group Capabilities for comparison
18	
Data Output Explain Messages Notifications	
	report character varying
1	
2	===== Groups =====
3	
4	Group = 3D UI = enable2020 Desc = PGHD Products HomePageWidth = 100
5	Group = Administrator UI = classic Desc = System Administration group HomePageWidth = 50
6	Group = Administrator UI = enable2020 Desc = System Administration group HomePageWidth = 50
7	Group = Basic User UI = classic Desc = HomePageWidth =
8	Group = Books UI = enable2020 Desc = Photo Books, Story Books, and Calendars HomePageWidth = 100
9	Group = Cards UI = classic Desc = Cards and Stationery HomePageWidth =
10	Group = Content Development Administrators UI = classic Desc = HomePageWidth =
11	Group = Content Development Users UI = classic Desc = HomePageWidth =
12	Group = Product Merchant Users UI = classic Desc = HomePageWidth =

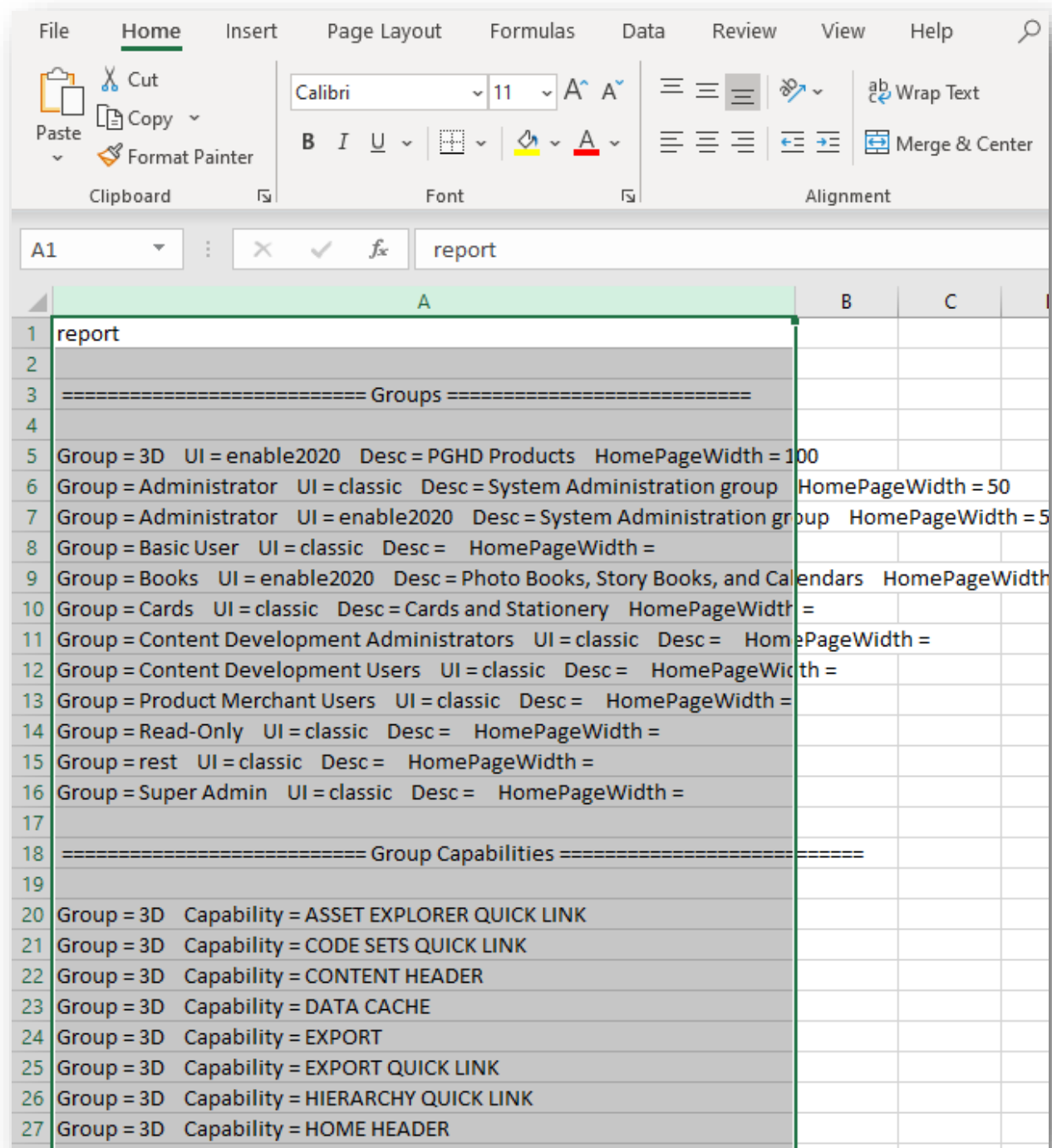
5. Click the Download as CSV Button:



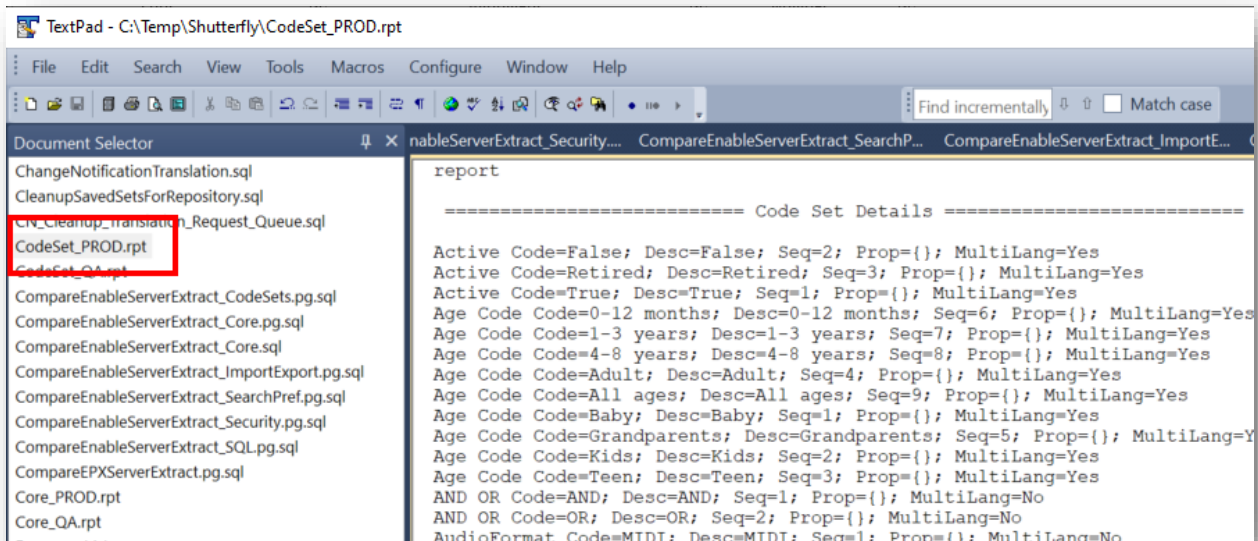
6. Open the resulting file in Excel:



7. Select Column A and copy its contents:



8. Paste the contents to a text editor:



9. If the PostgreSQL file is going to be compared to one extracted from SQL server, the PostgreSQL file will have double quotes around lines having a comma or double-quote character (and double-quotes will be escaped), so the extraneous double quotes need to be removed by performing the following steps:
  - a. Perform a "Regular Expression" search/replace with the Search:
 

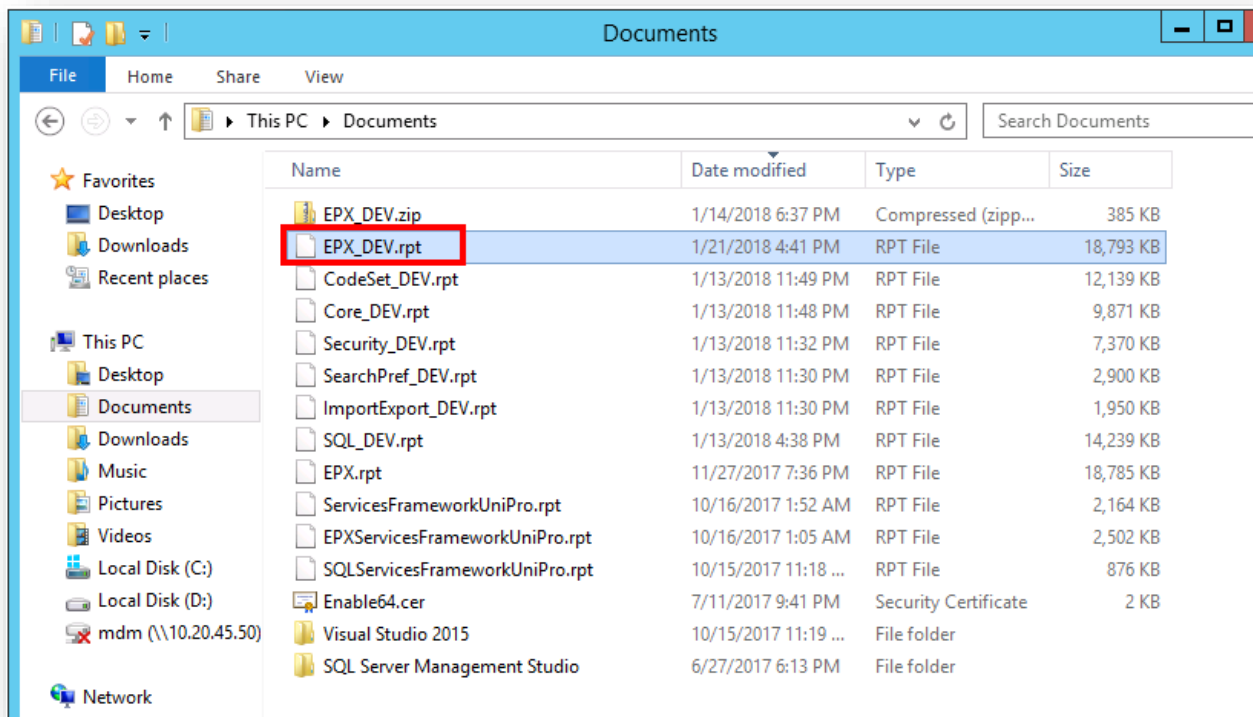
```
^\"
```

 And replace being blank.
  - b. Perform a "Regular Expression" search/replace with the Search:
 

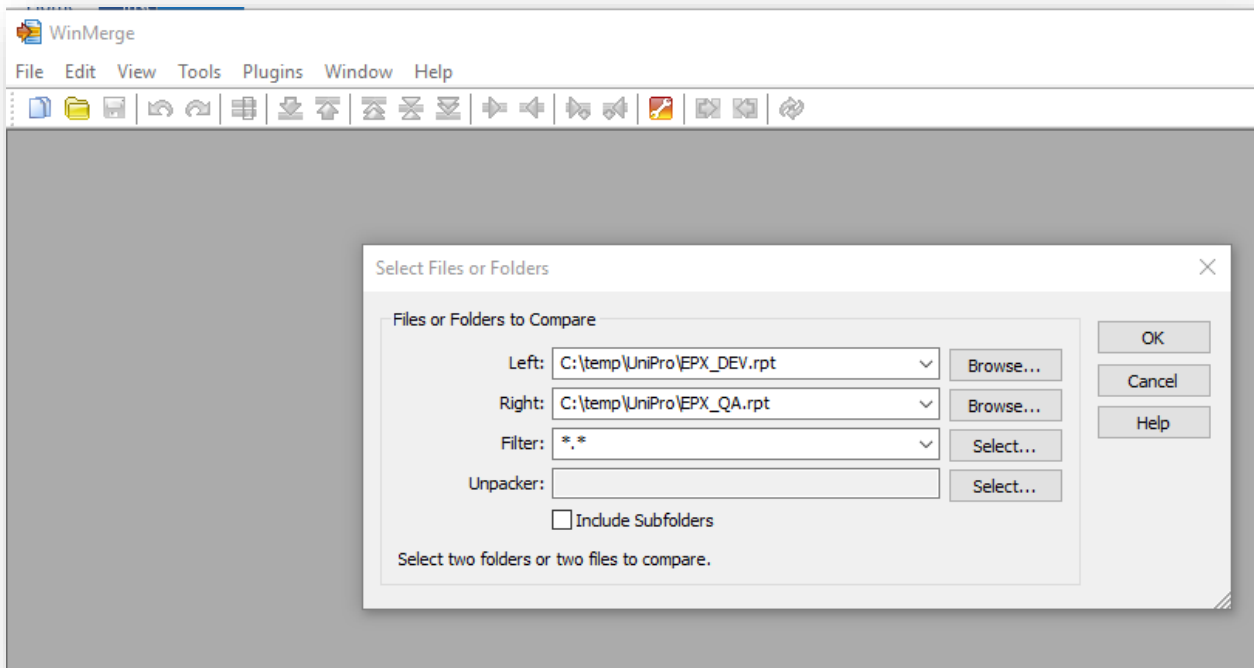
```
\"$
```

 And replace being blank.
  - c. Perform a search on "" and replace with " (be sure to only do one replace all operation on the file)
10. Save the text to a file using the recommended naming convention.
11. Copy the resulting .rpt file and use in comparisons and to update the source control (e.g., Subversion or GIT) if versioning the data model contents:





12. Repeat the above steps on the same database using the same script for the second environment.
13. Run the WinMerge utility to compare the two rpt files:



## Procedure - Generate Partial EPX Report

An extract that includes all workflows is most useful when comparing the EPX configuration between two environments (e.g., DEV vs. QA or PROD) and the generated rpt file can be fairly large. For facilitating the archiving of workflow-specific flows in the chosen source control platform, or for migration of a subset of workflows from one environment to another, having a generated file of a subset of the EPX objects can be more useful. The partial EPX report can be generated by performing the following steps:

1. Make a copy of the CompareEPXServerExtract.sql script and edit it.
2. Eliminate or separate the following sections to different file(s):

```
-- Extract Users for comparison
print '=====Users====='
```

```
-- Extract Roles for comparison
print '=====Roles====='
```

```
-- Extract Role Members for comparison
print '=====Role Members====='
```

3. Edit each of the remaining sections to include a new condition in the WHERE clause to limit the report to a specific set of workflows:
  - a. For the section:

```
-- Extract Process Flows for comparison
print '=====Process Flows====='
```

- b. Add the condition:

```
WHERE p.NAME in ('<workflow1>', '<workflow2>', '<workflow3>')
```

- c. After:

```
FROM [P_PROCESS] p
```

- d. For the section:

```
-- Extract Process Activities for comparison
print '=====Process Flow Activities====='
```

- e. Add the condition:

```
AND p.NAME in ('<workflow1>', '<workflow2>', '<workflow3>')
```

- f. After:

```
where a.DELETED_IND = 0
```

- g. For the section:

```
-- Extract Process Activity Viewers for comparison
print '=====Process Flow Activity Viewers====='
```

- h. Add the condition:

```
AND p.NAME in ('<workflow1>', '<workflow2>', '<workflow3>')
```

- i. After:

```
WHERE a.DELETED_IND = 0
```

- j. For the section:

```
-- Extract Process Flow Activity Transitions for comparison
print '=====Process Flow Activity Transitions====='
```

- k. Add the condition:

```
AND p.NAME in ('<workflow1>', '<workflow2>', '<workflow3>')
```

## I. After:

```
AND a2.DELETED_IND = 0
```

4. In each case, the condition IN clause should list the names of the process flows or subflows to be included in the report. For example, if a report is needed on the New Brand Approval workflow, the altered file would be:

```
-- Extract Process Flows for comparison
print '=====Process Flows=====
SELECT 'Flow=' + p.[NAME] as ProcessFlowName
, 'Desc=' + isnull(p.DESRIPTION, '') as ProcessFlowDescription
, case when (p.[PROCESS_TYPE_CODE] = 1) then 'Process Flow'
      when (p.[PROCESS_TYPE_CODE] = 2) then 'SubFlow'
      when (p.[PROCESS_TYPE_CODE] = 3) then 'Personal SubFlow'
      else CAST(p.[PROCESS_TYPE_CODE] as VARCHAR) end as FlowType
, 'FlowValid=' + case when (p.[VALID_IND] = 1) then 'Yes' else 'No' end as
Valid
FROM [P_PROCESS] p
WHERE p.NAME in ('New Brand Approval')
order by p.[NAME]

-- Extract Process Activities for comparison
print '=====Process Flow
Activities=====
SELECT '**** Flow=' + p.[NAME] as ProcessFlowName
      , 'Activity=' + a.NAME as ActivityName
      , 'Type=' + case when (a.ACTIVITY_TYPE_CODE = 1) then 'AUTOMATIC: ' +
a.ARC_ACTOR_NAME
      when (a.ACTIVITY_TYPE_CODE = 2) then 'SUBFLOW'
      when (a.ACTIVITY_TYPE_CODE = 3) then 'DECISION_POINT'
      when (a.ACTIVITY_TYPE_CODE = 4) then 'DISTRIBUTED_SUBFLOW'
      when (a.ACTIVITY_TYPE_CODE = 5) then 'WORK_ITEM_MERGE'
      when (a.ACTIVITY_TYPE_CODE = 6) then 'ITERATION'
      when (a.ACTIVITY_TYPE_CODE = 7) then 'JOIN: ' + case when (a.JOIN_TYPE = 1)
then 'OR' else 'AND' end
      when (a.ACTIVITY_TYPE_CODE = 8) then 'MANUAL'
      when (a.ACTIVITY_TYPE_CODE = 9) then 'ANONYMOUS'
      when (a.ACTIVITY_TYPE_CODE = 10) then 'PERSONAL_SUBFLOW'
      when (a.ACTIVITY_TYPE_CODE = 11) then 'SPLIT'
      when (a.ACTIVITY_TYPE_CODE = 12) then 'WORK_ITEM_REPEATER'
      when (a.ACTIVITY_TYPE_CODE = 14) then 'WORK_ITEM_PURGE'
      when (a.ACTIVITY_TYPE_CODE = 16) then 'ENDING_POINT'
      when (a.ACTIVITY_TYPE_CODE = 17) then 'LOAD_BALANCE'
      when (a.ACTIVITY_TYPE_CODE = 18) then 'CHANGE_PRIORITY'
      when (a.ACTIVITY_TYPE_CODE = 19) then 'SUBFLOW_EXIT'
      end as ActivityType
      , 'Enabled=' + case when (a.enabled_ind = 1) then 'E' else 'D' end as Enabled
      , 'Start=' + case when (a.start_point_ind = 1) then 'Y' else 'N' end as Start
      , 'End=' + case when (a.end_point_ind = 1) then 'Y' else 'N' end as [End]
```

```

        , 'SendOnError=' + case when (a.ERROR_SEND_IND = 1) then 'Y' else 'N' end as
[SendOnError]
        , 'Valid=' + case when (a.VALID_IND = 1) then 'Y' else 'N' end as
ActivityValid
        , 'Key=' + ap.PROPERTY_KEY + ' = ' +
            isnull(case when (ap.PROPERTY_KEY = 'lastSentDate') THEN ''
            else case when (ap.PROPERTY_VALUE is not null) then ap.PROPERTY_VALUE
                else isnull(convert(varchar(max),
convert(varbinary(max),ap.PROPERTY_VALUE_BLOB)), '')
            end
            end, '') as ActivityPropertyKey
FROM [P_PROCESS] p
join P_ACTIVITY a on p.PROCESS_ID = a.PROCESS_ID
left outer join P_ACTIVITY_PROPERTY ap on a.ACTIVITY_ID = ap.ACTIVITY_ID
where a.DELETED_IND = 0
AND p.NAME in ('New Brand Approval')
order by p.[NAME], a.NAME, ap.PROPERTY_KEY

```

```

-- Extract Process Activity Viewers for comparison
print '=====Process Flow Activity
Viewers=====
SELECT '**** Flow=' + p.[NAME] as ProcessFlowName
        , 'Activity=' + a.NAME as ActivityName
        , 'Start=' + case when (a.start_point_ind = 1) then 'Y' else 'N' end as
Start
        , 'Viewer=' + wiv.NAME as ViewerName
        , 'URL=' + wiv.URL as ViewerURL
        , 'Default=' + case when wiv.DEFAULT_IND = 0 then 'No'
            when wiv.DEFAULT_IND = 1 then 'Yes'
            else 'Unknown:' + CAST(wiv.default_ind as VARCHAR)
            end as DefaultViewer
FROM [P_PROCESS] p
join P_ACTIVITY a on p.PROCESS_ID = a.PROCESS_ID
join p_ACTIVITY_VIEWER av on av.ACTIVITY_ID = a.ACTIVITY_ID
join P_WORK_ITEM_VIEWER wiv on av.VIEWER_ID = wiv.VIEWER_ID
WHERE a.DELETED_IND = 0
AND p.NAME in ('New Brand Approval')
order by p.[NAME], a.NAME, wiv.NAME

```

```

-- Extract Process Flow Activity Transitions for comparison
print '=====Process Flow Activity
Transitions=====
SELECT 'Flow=' + p.[NAME] as ProcessFlowName
        , 'From=' + a.NAME as ActivityName
        , 'To=' + isnull(a2.NAME, '') as TargetActivityName
        , case when (tc.CONDITION_ID IS not null) then
            'Con=' + case when (c.CONDITION_TYPE_CODE = 3) then 'Otherwise'
            when (c.CONDITION_TYPE_CODE = 1) then 'Simple: ' + c.COMPARE_KEY +
C.OPERATOR_CODE + c.COMPARE_VALUE
            when (c.CONDITION_TYPE_CODE = 2) then 'Advanced: ' +
isnull(convert(nvarchar(max),c.FREE_EXPRESSION_CLOB), '')
            else 'Unknown: ' + cast(c.CONDITION_TYPE_CODE as varchar) end
            else '' end as Condition

```

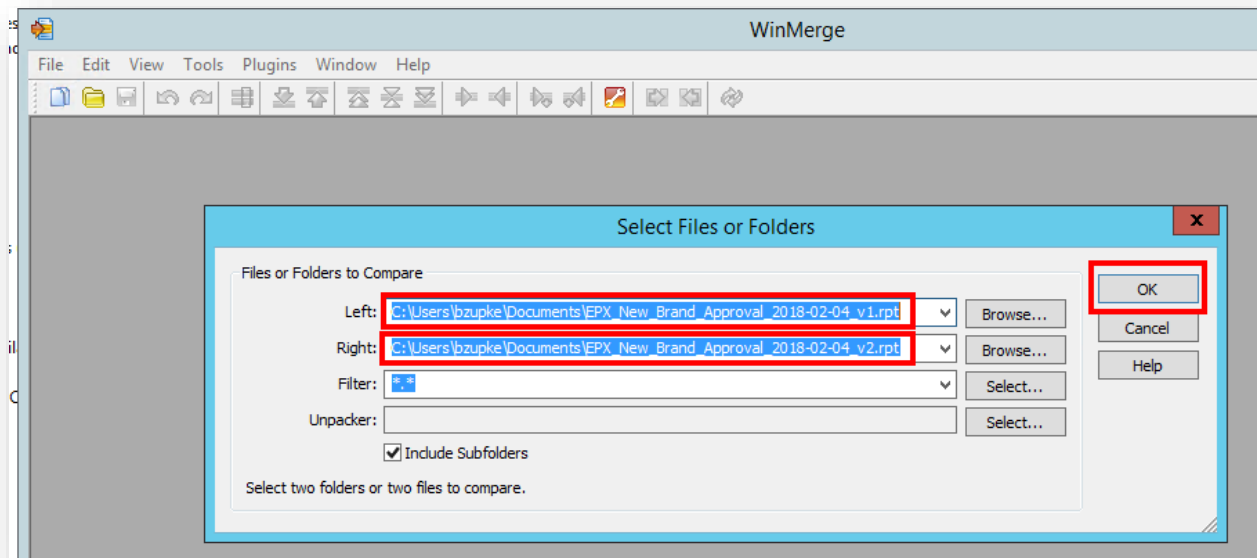
```
FROM [P_PROCESS] p
join P_ACTIVITY a on p.PROCESS_ID = a.PROCESS_ID
left outer join P_TRANSITION t on t.PARENT_ACTIVITY_ID = a.ACTIVITY_ID
left outer join P_ACTIVITY a2 on t.CHILD_ACTIVITY_ID = a2.ACTIVITY_ID
left outer join P_TRANSITION_CONDITION tc on tc.TRANSITION_ID = t.TRANSITION_ID
left outer join P_CONDITION c on tc.CONDITION_ID = c.CONDITION_ID
WHERE a.DELETED_IND = 0
AND a2.DELETED_IND = 0
AND p.NAME in ('New Brand Approval')
order by 1,2,3,4
```

5. Using the modified script, follow the steps in the Procedure – Generate Full Report (MS SQL Server/PostgreSQL)
6. Reports on individual process flows or subflows can be archived in the source control repository as separate artifacts. The native comparison tool or an external comparison tool such as WinMerge can be used to compare two versions of the same workflow.

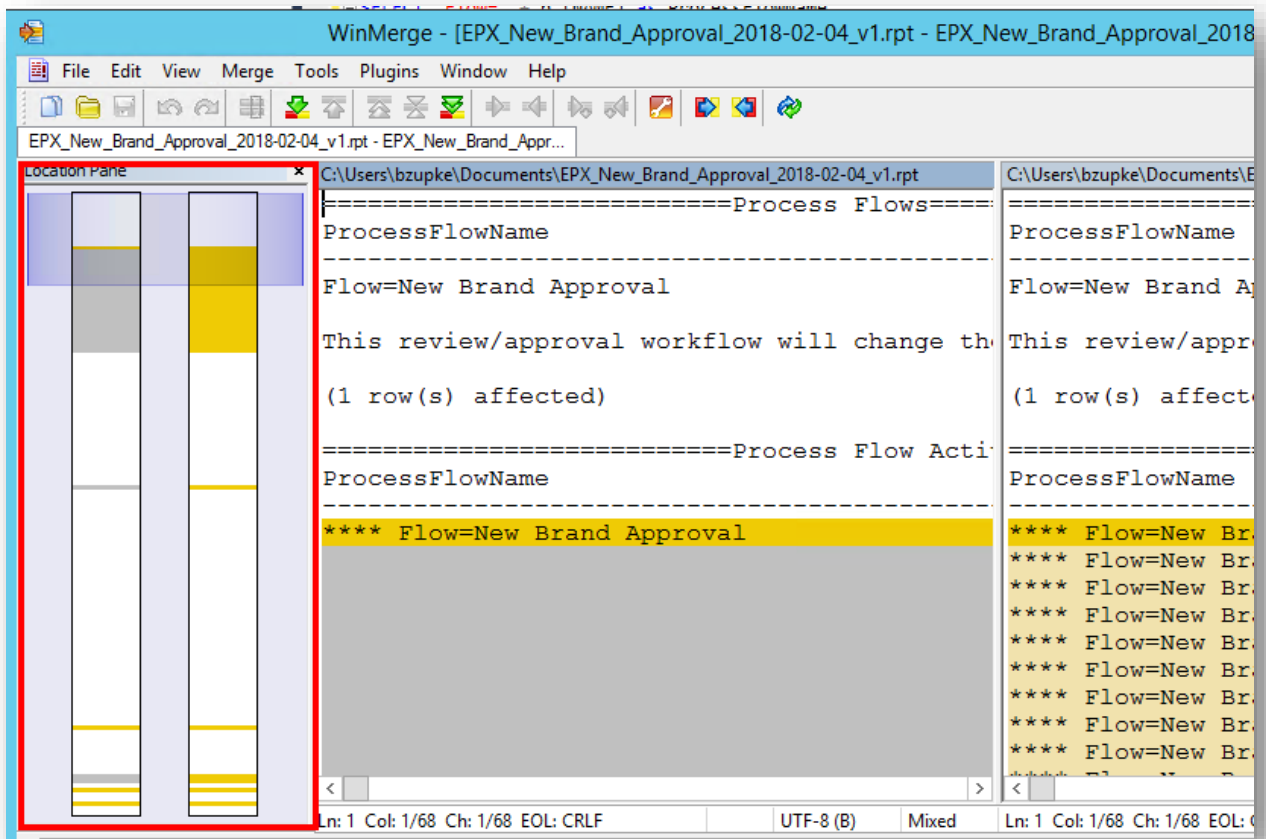
## Procedure – Comparing Extraction Reports Using WinMerge

Two reports generated using the same extraction SQL can be compared using WinMerge, which will highlight the differences between the two versions by performing the following steps:

1. Open WinMerge.
2. Select two versions of the report file generated by the same query in either two different systems or from the same system before and after changes are made:



3. The tool graphically highlights the sections of the two files that have differences on the left:



4. The tool graphically highlights the lines in the files that are different on the right:

- [illegible]