

# Contents

<b>Overview .....</b>	<b>3</b>
<b>Introduction .....</b>	<b>3</b>
<b>Features at a Glance .....</b>	<b>3</b>
<b>License .....</b>	<b>3</b>
<b>Legal Disclaimer .....</b>	<b>3</b>
<b>Found a bug .....</b>	<b>3</b>
<b>Installation .....</b>	<b>4</b>
<b>Install required package .....</b>	<b>4</b>
<b>Install plugin.....</b>	<b>5</b>
<b>Using in your application .....</b>	<b>6</b>
<b>Configuration Settings.....</b>	<b>8</b>
<b>Return Data.....</b>	<b>8</b>
<b>Maximum Rows .....</b>	<b>8</b>
<b>Download when click on (jQuery Selector).....</b>	<b>8</b>
<b>Custom Width .....</b>	<b>9</b>
<b>Installing sample application .....</b>	<b>9</b>
<b>FAQ (How To) .....</b>	<b>9</b>
<b>Replace default file name (Excel.xlsx) to the custom file name.....</b>	<b>9</b>
<b>Easy implement this functionality to all Interactive report in your application</b>	<b>9</b>
<b>Adding custom download button .....</b>	<b>11</b>

# GPV Interactive Report to MSExcel

---

How to exclude reports columns from export/make columns that are visible in Excel but not visible in Interactive Report .....11

How to set Maximum Rows property for certain pages when plugin functionality are implemented to all Interactive report in your application. ....12

How to Enable Download-XLS Icon .....14

# GPV Interactive Report to MSExcel

---

## Overview

### Introduction

The "GPV Interactive Report to Excel" APEX plugin provide simple export of IR-data into Microsoft Excel.

### Features at a Glance

These Interactive Reports features are supported:

- Correct export of data types (String/Date/Number)
- Filtering and Sorting
- Control Break
- Computations
- Aggregations
- Highlighting
- Auto adjustable Column Width
- Date/Number Formats
- Custom downloaded file name (see FAQ)
- Can be easy implemented for all Interactive reports in application (see FAQ)

### License

The "GPV Interactive Report to MSExcel" APEX plugin and IR\_TO\_XML, XML\_TO\_XSLX, IR\_TO\_MSEXCEL packages are currently available for use in all personal or commercial projects under both MIT and GPL licenses. This means that you can choose the license that best suits your project and use it accordingly. Both licenses have been included with this software.

### Legal Disclaimer

The program(s) and/or file(s) are supplied as is. The author disclaims all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The author assumes no liability for damages, direct or consequential, which may result from the use of these program(s) and/or file(s).

### Found a bug

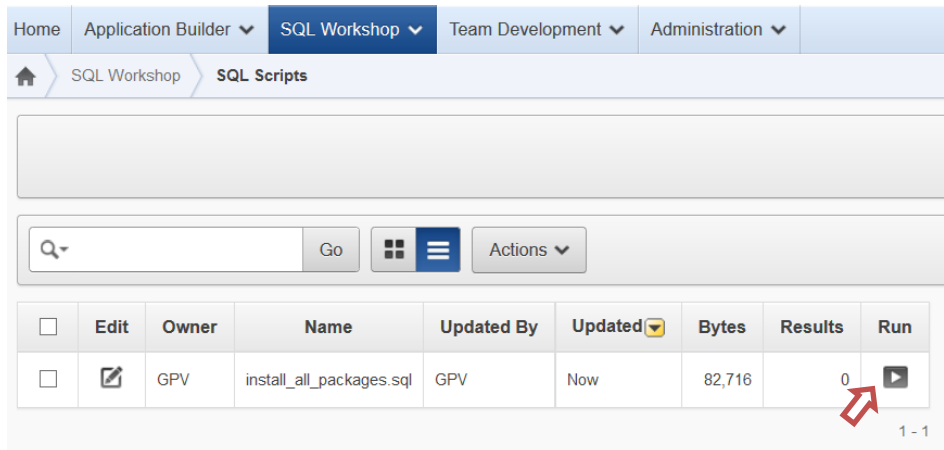
If you have found a bug, please send us debug information. To do this, set **Return Data** plugin options to **"Debug Information"**. Try to get your XML data again. You'll get text a file with debug information. Please send this file to [pavelglebov@outlook.com](mailto:pavelglebov@outlook.com). We'll try to fix this error as soon as possible.

### Installation

#### Install required package

This plugin requires IR\_TO\_XML, XML\_TO\_XSLX, IR\_TO\_MSEXCEL and AS\_ZIP packages. To install these packages, please use install\_all\_packages.sql all-in-one installation script.

To do this, navigate to the “SQL-Workshop > SQL-Scripts”. Upload file **install\_all\_packages.sql**.

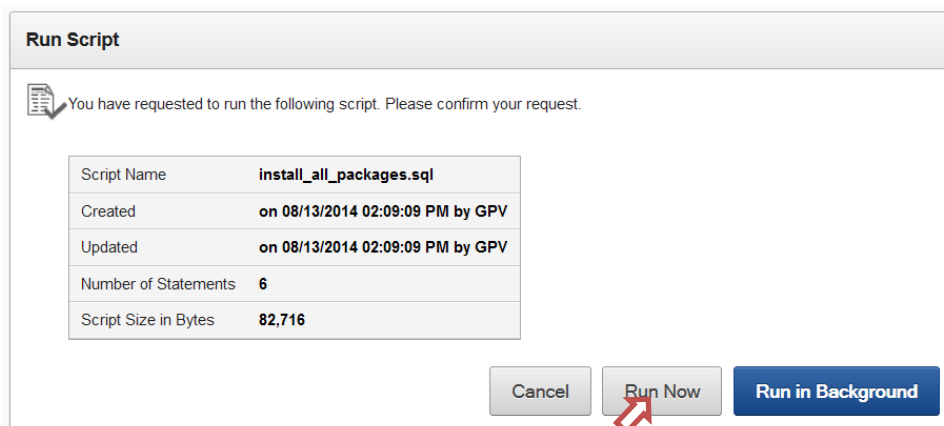


The screenshot shows the Oracle APEX SQL Workshop interface. The breadcrumb navigation is Home > Application Builder > SQL Workshop > SQL Scripts. Below the navigation is a search bar with a 'Go' button and an 'Actions' dropdown. A table lists the scripts:

<input type="checkbox"/>	Edit	Owner	Name	Updated By	Updated	Bytes	Results	Run
<input type="checkbox"/>		GPV	install_all_packages.sql	GPV	Now	82,716	0	

A red arrow points to the 'Run' icon in the last row. The page number '1 - 1' is visible at the bottom right.

This script will be created all required packages. Run this script by pressing Run-icon.



The 'Run Script' dialog box displays the following information:

Script Name	install_all_packages.sql
Created	on 08/13/2014 02:09:09 PM by GPV
Updated	on 08/13/2014 02:09:09 PM by GPV
Number of Statements	6
Script Size in Bytes	82,716

At the bottom, there are three buttons: 'Cancel', 'Run Now', and 'Run in Background'. A red arrow points to the 'Run Now' button.


Press “Run Now” button.

# GPV Interactive Report to MSExcel

HomeApplication BuilderSQL WorkshopTeam DevelopmentAdministration

SQL WorkshopSQL ScriptsManage Script Results

Q-GoActions

<input type="checkbox"/>	Script	Run By	Started	Elapsed	Status	Statements	Bytes	View Results
<input type="checkbox"/>	<a href="#">install_all_packages.sql</a>	GPV	2 minutes ago	1.06	Completed	6 of 6	0	

1 - 1

After execution take a look at execution results. At first open “Manage Script Results” tab by pressing on link in **View Results** column.

HomeApplication BuilderSQL WorkshopTeam DevelopmentAdministration

SQL WorkshopSQL ScriptsResults

Script: install\_all\_packages.sqlStatus: Complete

View: ☐ Detail ☒ SummaryRows: 15Go

Number	Elapsed	Statement	Feedback	Rows
1	0.63	CREATE OR REPLACE package ir_to_xml as --ver 1.4. --	Package created.	0
2	0.80	CREATE OR REPLACE package body ir_to_xml as subtype	Package Body created.	0
3	0.06	CREATE OR REPLACE package as_zip is /*****	Package created.	0
4	0.01	CREATE OR REPLACE package body as_zip is -- c_LOCAL_FILE_H	Package Body created.	0
5	0.05	CREATE OR REPLACE package xml_to_xslx -- ver 1.0. IS WIDTH	Package created.	0
6	0.35	CREATE OR REPLACE PACKAGE body XML_TO_XSLX IS STRING_HEIGH	Package Body created.	0
7	0.02	CREATE OR REPLACE package IR_TO_MSEXCEL as function get_x	Package created.	0
8	0.03	CREATE OR REPLACE package body IR_TO_MSEXCEL as XLS_DOW	Package Body created.	0

row(s) 1 - 8 of 8

Download

Statements Processed 8  
Successful 8  
With Errors 0

There should be no errors.

## Install plugin

Navigate to “Shared Components > Plug-ins” and click **Import**. [Herefrom](#) you can follow the menu to upload plugin-file “**process\_type\_plugin\_gpv\_ir\_xml.sql**” and to install the plug-in using the file above.

## GPV Interactive Report to MSExcel

### Using in your application

1. Create “On Load - Before Regions”-process “**Plug-Ins**” type.

Process Type Process Attributes Process Messages Process Conditions

Cancel Next >

Page: 1 - Plugin Demonstration

Select the category of the process you wish to create:

☐ PL/SQL ☐ Reset Pagination ☒ Plug-ins

☐ Session State ☐ Data Manipulation ☐ Web Services

Process Type Process Attributes Process Messages Process Conditions

< Cancel Next >

Page: 1 - Plugin Demonstration

Select Plug-in:

☒ GPV Interactive Report to XML

Show Unsupported: No

### Select “GPV Interactive Report to XML”.

Process Type Process Attributes Process Messages Process Conditions

< Cancel Next >

Page: 1 - Plugin Demonstration

\* Name: Get Excel

\* Sequence: 10

\* Point: On Load - Before Regions

\* Type: GPV Interactive Report to XML [Plug-in]

Give this process a name (for example “Get Excel”).  
Select execution “Point”: “On Load - Before Regions”

## GPV Interactive Report to MSEXcel

The screenshot shows a configuration wizard for 'GPV Interactive Report to MSEXcel'. At the top, a progress bar indicates five steps: 'Process Type', 'Process Attributes', 'Process' (current step), 'Messages', and 'Process Conditions'. The 'Process' step is highlighted with a blue circle. Below the progress bar, the wizard has a header bar with a back arrow, a 'Cancel' button, a 'Create Process' button (highlighted with a red arrow), and a 'Next >' button. The main content area is titled 'Page: 1 - Plugin Demonstration' and 'Point: On Load - Before Regions'. It contains several configuration options: 'Return Data' is set to 'Generated XML-data'; 'Include Page Items' is set to 'Yes'; 'List of Items to Include' is an empty text field with a help icon; 'APEX Collection Name' is an empty text field; '\* Maximum Rows' is set to '1000'; and '\*try again\* message' is set to 'Report refreshed. Please try download again.'. At the bottom, there is a section titled '> Page Items'.

Press **“Create Process”**.

### Configuration Settings

#### Return Data

Choose "Excel XLSX" to return result of Interactive Report as MS-Excel file.

Choose "XML (Debug)" to return result of Interactive Report in XML format.

Choose "Debug TXT" to view debug information.

#### Maximum Rows

Rows greater than this value will not be exported.

To export all the rows set the value to 1000000000.

When empty - value from Interactive Report Attributes-> Maximum Row Count will be used.

Home Application Builder SQL Workshop Team Development Administration

Application Builder Application 26482 Page 1 Interactive Report Attributes

Home Region Definition Report Attributes Saved Reports Print Attributes

Region Name: Sample Interactive Report

**Pagination**

Pagination Type: Row Ranges X to Y

Pagination Display Position: Bottom - Right

Show Null Values as: -

Maximum Row Count: 1000

When more than maximum row data found message

The maximum row count for this report is #MAX\_ROW\_COUNT# rows. Please apply a filter to reduce the number of records in your query.

#### Download when click on (jQuery Selector)

Download starts when clicking on objects that are selected with this JQuery selector.

#### Replace standard IR XLS download

Replace standard IR download XLS menu functionality to download XLSX feature.

Download

Choose report download format:

CSV HTML Email XLS

Cancel

This functionality will be replaced



## GPV Interactive Report to MSeExcel

### Custom Width

Comma-delimited string with values, each value consist of column name and column width.

So, this string

PROJECT=151,TASK\_NAME=319,START\_DATE=133,

sets

width = 151 to the column with name = "PROJECT",

width = 319 to the column with name = "NAME",

width = 133 to the column with name = "START\_DATE"

**Please do not forgive comma at the end of the string!**

Width are set in special units that are **approximately** equal to pixels.

### Installing sample application

You can find sample application in SAMPLE\_APPLICATION folder in plugin archive.

To install sample application please follow this [Instructions](#).

### FAQ (How To)

#### Replace default file name (Excel.xlsx) to the custom file name

When not empty, value from *Report Attributes-> Download-> Filename* field will be used.

The screenshot shows the Oracle APEX 'Interactive Report Attributes' page. The breadcrumb trail is: Home > Application Builder > Application 26482 > Page 1 > Interactive Report Attributes. The 'Report Attributes' tab is selected. Below the tabs, there is a 'Download' section. In the 'Download Formats' section, the following options are checked: CSV, HTML, Email, and XLS. The 'Filename' field is highlighted with an orange border and contains the text 'REPORT-&SYSDATE'. An orange arrow points from the 'Report Attributes' tab to the 'Download' section.

#### Easy implement this functionality to all Interactive report in your application

1. Install plugin and required packages.
2. Create "On Load: After Header" application process.

## GPV Interactive Report to MSeExcel

Home Application Builder SQL Workshop Team Development Administration

Application Builder Application 26482 Shared Components Application Processes Page 1

Application Processes History

Q- Go Actions Reset Create

No data found.

Identification Source Conditionality

Cancel Next

Application Processes run PL/SQL logic at specific points for each page in an application or as defined by the conditions under which they are set to fire. Note that "On Demand" processes fire only when called from specific pages.

Application: 26482 Sample Application

\* Name: Get Excel from IR

\* Sequence: 1

\* Point: On Load: After Header (page template header)

Type: PL/SQL Anonymous Block

"On Demand" processes should be declared at the page level whenever possible.

Identification Source Conditionality

Cancel Next

Application: 26482 Sample Application

Type: PL/SQL Anonymous Block

Point: On Load: After Header (page template header)

\* Process Text

```
get_excel_from_ir_ext(p_maximum_rows => null,
    p_jquery_selector => null,
    p_download_type => 'E', -- E -> Excel XLSX, X -> XML (Debug), T -> Debug TXT
    p_replace_xls => 'Y', --Y/N
    p_custom_width => null
);
```

Please set your custom parameters or leave default settings.

```
get_excel_from_ir_ext(
    p_maximum_rows => null,
    p_jquery_selector => null,
    p_download_type => 'E', -- E -> Excel XLSX,
    -- X -> XML (Debug),
    -- T -> Debug TXT
    p_replace_xls => 'Y', --Y/N
    p_custom_width => null
);
```

Parameters description:

- p\_maximum\_rows - see *Maximum Rows* plugin parameter
- p\_jquery\_selector - see *Download when click on (jQuery Selector)* plugin parameter
- p\_download\_type - see *Return Data* plugin parameter
- p\_replace\_xls - see *Replace standard IR XLS download* plugin parameter
- p\_custom\_width - see FAQ

## GPV Interactive Report to MSExcel

When you want to exclude this functionality from couple of pages, use **Conditions**.

**Conditions**

Condition Type  
Current Page Is NOT in Expression 1 (comma delimited list of pages) ▼  
[PL/SQL] [item / column=value] [item / column not null] [item / column null] [request=e1] [page in] [page not in] [exists] [never] [none]

Expression 1  
10,15,20

### Adding custom download button

1. First create a button with certain **Static ID** and Action “**Defined by Dynamic Actions**”

**Attributes**

Static ID GETEXCEL

Button Style Template Based Button ▼

\* Button Template Interactive Report Button ▼

Button Type Normal ▼

Button CSS Classes

Button Attributes

**Action When Button Clicked**

Action Defined by Dynamic Action ▼

Execute Validations Yes ▼

Database Action - No Database Action - ▼

2. Edit “Download when click on (jQuery Selector)” property of Process with type “GPV Interactive Report to MSExcel [Plug-in]”. Add **Static ID** of the button with ‘#’-prefix.

**Settings**

Replace standard IR XLS download Yes ▼

Maximum Rows 500

Download when click on (jQuery Selector) #GETEXCEL

Download Type Excel XLSX ▼

**How to exclude reports columns from export/make columns that are visible in Excel but not visible in Interactive Report**

## GPV Interactive Report to MSEXcel

Use **Conditional Display** column property for this.

**Conditional Display**

Condition Type

PL/SQL Expression

[PL/SQL] [item / column=value] [item / column not null] [item / column null] [request=e1] [page in] [page not in] [exists] [never] [none]

Expression 1

nv1(:REQUEST, 'N') not like 'GPV\_IR\_TO\_MSEXCEL%'

☐ Do not validate code (parse code at runtime only).

Use `nv1(:REQUEST, 'N') not like 'GPV_IR_TO_MSEXCEL%'` Expression to exclude column from export, and `nv1(:REQUEST, 'N') like 'GPV_IR_TO_MSEXCEL%'` Expression to make column visible on export only.

### How to set Maximum Rows property for certain pages when plugin functionality are implemented to all Interactive report in your application.

! This hint references to *“Easy implement this functionality to all Interactive report in your application”* paragraph. Please read this paragraph first.

Typical task is to get users with special rights an ability to download unlimited data from Interactive Reports on certain pages.

If you use **“On Load: After Header”** application process to use functionality of this plugin on all Interactive report, parameter `p_maximum_rows` is the same for all pages in application. You need to find easiest way to modify this parameter for certain page. First you need to create protected hidden items on this pages. This items should have specific names (`P23_DOWNLOAD_MAX_ROWS` for page 23) and contains count of maximal downloaded rows depending on user rights (10000000000 for unlimited). To fill this items you need to create **“After Header”** calculation process.

## GPV Interactive Report to MSeExcel

For example:

Item Name	
Page:	23 Contact Search
* Item Name	<input type="text" value="P23_DOWNLOAD_MAX_ROWS"/>
Type	<input type="text" value="PL/SQL Function Body"/>
Computation Point	
* Sequence	<input type="text" value="30"/>
Computation Point	<input type="text" value="Before Header"/>
Source	
* Computation	
<pre>if app_rights.has_role('UNLIM_CONTACT_EXPORT') = app_rights.YES then   return 100000000; else   return null; end if;</pre>	

This calculation process gives limit of 100000000 rows users with **UNLIM\_CONTACT\_EXPORT** role, and set default value for others.

Now you need modify your “On Load: After Header” application process to use this variable automatically when it exist on page. Use a little hack: use substitution string in item name.

```
get_xlsx_from_ir_ext(
  p_maximum_rows => null,
  p_maximum_rows => :P&APP_PAGE_ID._DOWNLOAD_MAX_ROWS,
  p_jquery_selector => null,
  p_download_type => 'E', -- E -> Excel XLSX
  p_replace_xls => 'Y', --Y/N
  p_custom_width => null
);
```

Now, when current page is 23 and page item P23\_DOWNLOAD\_MAX\_ROWS exist, it will be used. Otherwise will be used default value, Maximum Row Count of current Interactive Report.

Maximum Row Count	<input type="text" value="1000"/>
When more than maximum row data found message	
<pre>The maximum row count for this report is #MAX_ROW_COUNT# rows. Please apply a filter to reduce the number of records in your query.</pre>	

### How to Enable Download-XLS Icon .

1. Access the Administration Services home page (see [http://docs.oracle.com/cd/E11882\\_01/appdev.112/e12512/adm\\_login.htm#AEADM168](http://docs.oracle.com/cd/E11882_01/appdev.112/e12512/adm_login.htm#AEADM168))
2. Select Manage Service
3. Select Instance Settings, under Manage Environment Settings
4. Click Report Printing to focus on just the Report Printing attributes
5. Now, specify the following attributes: Oracle BI Publisher: Advanced Support
6. Apply Changes.

**! You dot'need installed BI Publisher, you just need to set this setting.**