

Focal Point[®]

Custom Chart Plugin Reference Manual

7.42

Publication information

FPNC-7420-00 (November 2020)

Information in this publication is subject to change. Changes will be published in new editions or technical newsletters.

Documentation set

The documentation relating to this product includes:

- Focal Point APM Workspace Deployment Guide
- Focal Point APM Workspace User Guide
- Focal Point Business Rule Development and Usage Guide
- Focal Point Custom Chart Plugin Reference Manual
- Focal Point REST RDF API Reference Manual
- Focal Point RESTful API Reference Manual
- Focal Point Web Services API Reference Manual

Copyright notice

Focal Point® (the Programs and associated materials) is a proprietary product of UNICOM Systems, Inc. – a division of UNICOM Global. The Programs have been provided pursuant to License Agreement containing restrictions on their use. The programs and associated materials contain valuable trade secrets and proprietary information of UNICOM Systems, Inc. and are protected by United States Federal and non-United States copyright laws. The Programs and associated materials may not be reproduced, copied, changed, stored, disclosed to third parties, and distributed in any form or media (including but not limited to copies on magnetic media) without the express prior written permission of UNICOM Systems, Inc., UNICOM Plaza Suite 310, 15535 San Fernando Mission Blvd., Mission Hills, CA 91345 USA.

Focal Point®

© Copyright 1997-2020 All Rights Reserved. UNICOM Systems, Inc. – a division of UNICOM Global.

No part of this Program may be reproduced in any form or by electronic means, including the use of information storage and retrieval systems, without the express prior written consent and authorization of UNICOM Systems, Inc.

No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, without prior written permission from UNICOM Systems, Inc.

Disclaimer

We cannot guarantee freedom from, or assume any responsibility or liability for technical inaccuracies or typographical errors. The information herein is furnished for informational use only and should not be construed as a commitment by UNICOM Systems, Inc. – a division of UNICOM Global.

Trademarks

The following are trademarks or registered trademarks of UNICOM Systems, Inc. in the United States and/or other jurisdictions worldwide: Focal Point, UNICOM, UNICOM Systems.

Trademark acknowledgements

Divisions of UNICOM Global:

DETEC, iET Solutions, Macro 4, SoftLanding, UNICOM.

IBM:

IBM®, AIX®, CICS®, CICS/ESA®, CICS TS® CMAC®, Db2®, DFSMS/MVS®, Domino®, DOORS®, ESCON®, IBM MQ, IBM Z®, IMS™, iSeries®, Jazz™, Language Environment®, LE®, Lotus®, MQSeries®, Multi-Factor Authentication for z/OS®, MVS™, MVS/ESA®, OMEGAMON®, OS/390®, OS/400®, Power®, POWER®, pSeries®, RACF®, Rational®, Rational Team Concert®, RMF™, S/370®, S/390®, SMF®, System/390®, System i®, System p®, System z®, VisualAge®, VM/ESA®, VSE/ESA®, VTAM®, WebSphere®, z/OS®, z/VM®, z/VSE®, zSeries®, z Systems® and the IBM logo are trademarks or registered trademarks of IBM Corporation in the United States or other countries or both.

Microsoft:

Active Directory, Excel, Internet Explorer, Microsoft, Notepad, PowerPoint, Visio, Visual Basic, Windows, Windows 2000, Windows NT, Windows Server 2003, Windows Server 2007, Windows Vista, Windows XP, WordPad and/or other Microsoft products referenced are either trademarks or registered trademarks of Microsoft Corporation.

Adobe Systems Incorporated:

Adobe®, the Adobe logo, Acrobat® and Adobe Reader® are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Amazon.com, Inc.:

Amazon Web Services, the "Powered by AWS" logo and Amazon S3 are trademarks of Amazon.com, Inc. or its affiliates in the United States and/or other countries.

Apache Software Foundation:

Apache, Apache Commons, Apache Tomcat, Log4j, Lucene, Solr, Tomcat, Xalan and Xerces are trademarks of the Apache Software Foundation.

Apple Inc.:

AirPrint, App Store, iPad, iPhone and Safari are trademarks, registered trademarks or service marks of Apple Inc. registered in the United States and other countries.

Atlassian:

Atlassian and Jira are registered trademarks of Atlassian.

AXELOS Limited:

ITIL® is a registered trademark of AXELOS Limited, used under permission of AXELOS Limited.

babel:

A GitHub repository with contributions from Sebastian McKenzie and others and distributed under the MIT license.

BEA Systems, Inc.:

JRockit and WebLogic are registered trademarks of BEA Systems, Inc.

BCD Software, LLC:

ProGen Plus and WebSmart are trademarks or registered trademarks of BCD Software, LLC.

BMC Software Inc.:

Boole & Babbage, Data Packer, Optimizer and Super Optimizer are trademarks or registered trademarks of BMC Software, Inc., or its affiliates or subsidiaries (collectively, "BMC Software").

BSD:

PostgreSQL is distributed under the classic BSD license. (Portions Copyright © 1996-2006, PostgreSQL Global Development Group; Portions Copyright © 1994-1996 Regents of the University of California.)

CA, Inc.:

CA 2E, CA ACF2, CA Datacom, CA Endevor, CA IDMS, CA InterTest, CA NetMaster, CA Optimizer, CA Panexec, CA Panvalet, CA Ramis, CA Telon and CA Top Secret are registered trademarks of CA, Inc.

Candescent SoftBase LLC:

SoftBase is a registered trademark of Candescent SoftBase LLC.

Canonical Ltd:

Ubuntu is a registered trademark of Canonical Ltd.

Cappgemini:

IAF is a trademark of Cappgemini.

Chicago-Soft, Ltd.:

QuickRef is a trademark of Chicago-Soft, Ltd.

Cincom Systems, Inc.:

MANTIS is a registered trademark of Cincom Systems, Inc.

Cisco Systems, Inc.:

Cisco, Cisco Systems, Cisco Unity Express, the Cisco logo, the Cisco Systems logo and IOS are trademarks or registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

Computer Sciences Corporation:

Hogan and Hogan Umbrella are trademarks or registered trademarks of Computer Sciences Corporation.

Compuware Corporation:

Abend-AID and Compuware are trademarks or registered trademarks of Compuware Corporation.

Dell Inc.:

Dell and the Dell logo are trademarks of Dell Inc.

Emtex Limited:

Emtex and VIP are trademarks of Emtex Limited.

ExcelSystems Software Development Inc.:

WebSmart is a registered trademark of ExcelSystems Software Development Inc.

FasterXML, LLC:

Jackson is a FasterXML, LLC GitHub repository and is released under the terms of the Apache License 2.0.

Jean-loup Gailly and Mark Adler:

zlib is a registered trademark or trademark of Jean-loup Gailly and Mark Adler.

GNU General Public License:

Cygwin is free software released under the GNU General Public License.

Google Inc.:

Android™ platform, Google® and Google Chrome® are trademarks or registered trademarks of Google Inc.

Hewlett-Packard Development Company, L.P.:

HP and HP-UX are registered trademarks of Hewlett-Packard Development Company, L.P., and/or its subsidiaries.

hjson:

A GitHub repository with contributions from Christian Zangl and others and distributed under the MIT license.

Idera, Inc.:

LANSA is a registered trademark of Idera, Inc.

IDM Computer Solutions, Inc.:

UltraEdit is a trademark owned by IDM Computer Solutions, Inc. All rights reserved.

Infor (US), Inc.:

Lawson is a registered trademark of Infor (US), Inc.

Innovation Data Processing:

IAM is a registered trademark of Innovation Data Processing Corporation.

Hilton Janfield:

jquery.enhsplitter is a GitHub repository owned by Hilton Janfield and is released under the terms of the GNU Lesser General Public License.

Kofax, Inc.:

Kofax, the Kofax logo and Kofax Capture are the trademarks or registered trademarks of Kofax, Inc., in the United States and other countries.

Daniel Lovasko:

libdes © Copyright 2017-2018 Daniel Lovasko All rights reserved.

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR 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 THE AUTHOR OR CONTRIBUTORS 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.

Massachusetts Institute of Technology (MIT):
Kerberos is a trademark of the Massachusetts Institute of Technology (MIT).

Merrill Pty Ltd.:
MXG is a registered trademark of Merrill Pty Ltd.

Mozilla Foundation:
Firefox is a registered trademark of the Mozilla Foundation.

Mozilla Public License:
Expat is free software released under the Mozilla Public License.

OASIS:
BPEL is a trademark of OASIS.

Object Management Group:
BPMN™, Object Management Group®, OMG® and UML® are trademarks or registered trademarks of Object Management Group.

The Open Group:
ArchiMate, The Open Group, TOGAF and UNIX are registered trademarks of The Open Group in the United States and other countries.

Oracle Corporation:
EJB, Java, JDBC, JDK, JMX, JRE, JSP, JVM, OpenJDK, Solaris and SunOS are trademarks or registered trademarks of Oracle Corporation and/or its affiliates. Oracle is a registered trademark, and other Oracle product names, service names, slogans or logos are trademarks or registered trademarks of Oracle Corporation.

The pgAdmin Development Team:
pgAdmin is a trademark of The pgAdmin Development Team.

Pink Elephant Inc.:
Pink Elephant and its logo, PinkVERIFY, PinkSCAN, PinkATLAS, PinkSELECT and PinkREADY are either trademarks or registered trademarks of Pink Elephant Inc.

Python Software Foundation:
Python is a registered trademark of the Python Software Foundation in the United States.

QOS.ch:
SLF4J is from QOS.ch and its source code and binaries are distributed under the MIT license.

Red Hat, Inc.:
Red Hat, Red Hat Enterprise Linux, the Shadowman logo and JBoss are registered trademarks of Red Hat, Inc. in the United States and other countries.

SAP AG:
SAP, the SAP logo, the SAP Partner logo, Duet, SAP ArchiveLink, SAP Crystal Reports, SAP NetWeaver, SAP R/3 and SAPHIRE are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

SAS Institute Inc.:
SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.

Secretary of State for Defence:
MODAF is a registered trademark of the Secretary of State for Defence.

Software AG:
Adabas and Natural are registered trademarks of Software AG. Software AG and all Software AG products are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc.

SPARC International, Inc.:
SPARC is a registered trademark of SPARC International, Inc. (Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.)

Splunk Inc.
Splunk®, Splunk>®, Listen to Your Data®, The Engine for Machine Data®, Hunk®, Splunk Cloud®, Splunk® Light, SPL™ and Splunk MINT™ are trademarks or registered trademarks of Splunk Inc. in the United States and other countries.

Standardware Inc.:
COPE is a trademark of Standardware Inc.

Sun Microsystems, Inc.:
Sun, Sun Microsystems, the Sun logo, MySQL and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. or its subsidiaries in the United States and other countries.

SUSE:
openSUSE, the openSUSE logo, SUSE, the SUSE logo, YAST, and the Chameleon mark are registered trademarks of SUSE or its subsidiaries or affiliates. AutoBuild, SUSE Studio, SUSECON, "We adapt. You succeed." are trademarks of SUSE or its subsidiaries or affiliates.

Syncsort Inc.:
Syncsort is a registered trademark of Syncsort Inc.

Simon Tatham:
PuTTY is copyright Simon Tatham.

TIBCO Software Inc.:
JasperReports and Jaspersoft are registered trademarks of TIBCO Software Inc. and/or its subsidiaries in the United States and/or other countries.

Linus Torvalds:
Linux is a registered trademark of Linus Torvalds.

Unicode, Inc.:
Unicode and the Unicode logo are registered trademarks of Unicode, Inc. in the United States and other countries.

VMware, Inc.:
spring is a registered trademark of VMware, Inc.

Wireshark Foundation:
Wireshark and the "fin" logo are registered trademarks of the Wireshark Foundation.

XEROX CORPORATION:
XEROX, The Document Company and the stylized X are trademarks of XEROX CORPORATION.

X.Org Foundation:
X Window System is a trademark of the X.Org Foundation.

Zachman International, Inc.:
Zachman®, Zachman Framework™ and Zachman International® are trademarks or registered trademarks of Zachman International, Inc.

Additional trademarks and registered trademarks are the property of their respective owners.

Contents

1 Focal Point® Custom Chart Plugin Reference	7
Introduction	7
Overview Diagram	7
Configuration	8
Examples	8
Steps to create a script for Chart Plugin	8
Step 1 Create a base script	8
Step 2 Create an add-on script (optional step)	9
Step 3 Add CSS, JPEG or JS Files	9
Step 4 Create the Plugin	10
Steps to upload a Custom Chart Plugin	10
Step 1 Go to the Configure Plugin page	10
Step 2 Download the sample Chart Policy	11
Step 3 Unzip the file SampleChartPolicy.zip	11
Step 4 Import the sample workspace (optional step)	12
Step 5 Upload the Plugins	12
Step 6 Configure the plugin on the home page	14
Step 7 Configure the plugin on the Homepage for XSLT plugin	16
Step 8 Configure the plugin on a view page	17
CUSDATA/DAO object	18
CUSDATA API Information	19
getAttribIdFromName (attribName) > {String}	19
getAttribNameFromId (attribId) > {String}	19
getAttrNamesFromDAO (types) > {Array}	19
getBasicFpDataUrl () > {String}	20
getCurrentPluginId () > {String}	20
getCustomFpDataUrl (ATTR, conditions) > {String}	20
getData (ATTR, conditions, callback) > {Void}	21
getFPRestData (URL, callback, outputFormat) > {Void}	21
getPref () > {Object}	22
getPrefData (callback) > {Void}	22
postPrefData (callback) > {Void}	22
resize () > {Void}	23
savePref (JsonSettingsData) > {Void}	23

transform(xmlData, userDefinedXslFilePath) > {Void}	23
Data types	24
ID	24
Workspace	24
View	24
Element	24
Attribute	24
AttributeValue	25
AttributeSetup	26
ElementSet	27
AttributeType	27
AttributeSetupInfo	27
ChoiceSetup	27
ChoiceSetupItem	27
ViewInfo	27
HistoryEntry	28

1 Focal Point® Custom Chart Plugin Reference

Introduction

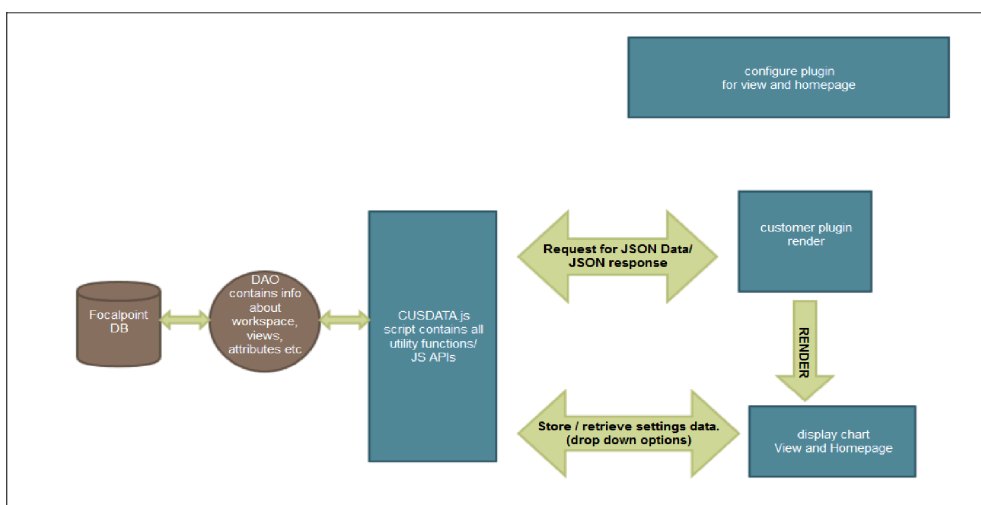
Data visualization capabilities that are included in Focal Point® help you to make value-based decisions in the field of Portfolio Management and Product Management. You can also use the 'Custom Chart Plugin' capability to develop and add your own unique charting/reporting capabilities as plugins.

This document describes the API and the Usage guide of 'Custom Chart Plugin' for Focal Point® version 7.1.2 onwards.

The Focal Point® Chart plugin is based on the Focal Point® REST services. The purpose is to provide access to Focal Point® data and services and facilitate the display of the data in the form of visualization charts and reports on home pages and view pages.

To use the Focal Chart plugin and understand this document, you need to be familiar with the Focal Point® REST Services, Java Script and JSON/XML data standards and have basic understanding of the Focal Point® data model, usage, and administration.

Overview Diagram



Configuration

As an Admin, you can upload the Custom Chart plugin in the **Applications > Plugins** settings and then configure homepage settings and view settings.

Charts and reports can be created by using any of the freely available data visualization libraries such as D3, Dimple, Plotly and XSLT.

The Focal Point® JavaScript API, which is provided for the development of the Custom Chart plugin, provides access to the following services:

- Retrieve attribute names and types available in a view
- Retrieve data in JSON format using the Focal Point® REST API
- Retrieve user settings from Focal Point® and save back to database

Examples

Code examples for Custom Chart plugins in Focal Point® can be found in the download link provided in **Administration > Application > Plugin** page.

The examples provided can be used as the basis for custom chart development.

Steps to create a script for Chart Plugin

Step 1 Create a base script

The path to the script has two parts; one is fixed and the other is a relative path to the script file.

The Fixed PATH (which will be the same for all plugins):

- For homepage plugins: `fpone/plugins/homepage/`
- For view plugins: `fpone/plugins/view/`

The Relative PATH (which will be different for each plugin) to script is:

```
<DIR PATH>/<scriptfilename>
```

The Complete PATH will be the Fixed PATH concatenated with the Relative PATH.

For example:

```
fpone/plugins/view/MYDIR/main
```

The base script file (`main.js`) would have the following structure.


```
define("fpone/plugins/view/MYDIR/main", [
    "dojo/ready"
    ], function(ready) {
function main(CUSDATA) {
}
    return main;
});
```

Step 2 Create an add-on script (optional step)

The following step is optional, and is only required if you want to add another script to the base script.

First create a script in AMD format which would look like the following example.

Note For view type plugin, the PATH would be: fpone/plugins/view/MYDIR/lib

The Relative PATH to the script is: MYDIR/lib/addon.js

```
define("fpone/plugins/view/MYDIR/lib", [], function() {
var Refresh = {};
//User defined functions
Refresh.fullDoc = function(target) {
    //user code
}
    return Refresh;
});
```

Next, to add this script to the base script, modify the base script file (`main.js`) to have the following structure.

```
define("fpone/plugins/homepage/MYDIR/main", [
    "dojo/ready",
    "fpone/plugins/homepage/MYDIR/lib/addon"
    ], function(ready, Refresh) {

function main(CUSDATA) {
    //Call your methods of addon
    //For eg
    Refresh.fullDoc();
}
    return main;
});
```

Step 3 Add CSS, JPEG or JS Files

To add CSS, JPEG or JS Files, create a directory structure to keep these files.

For example:

```
MYDIR/LIB/  
MYDIR/CSS/main.css  
MYDIR/IMG/clip.png  
MYDIR/JS/chart.js
```

In the HTML files just replace the `pluginBasePath` available with the `CUSDATA` object.

If you are using XSL files, use the following method:

Use the following replacement text:

For namespace URL use `@@FPNAMESPACE@@`

For example:

```
xmlns:ns="@@FPNAMESPACE@@"
```

For path to custom script files use `@@BASEPATH@@`

For example:

```
<link rel="stylesheet" type="text/css" href="@@BASEPATH@@XSLT/css/util.css"  
>
```

```
<script type="text/javascript"  
src="@@BASEPATH@@XSLT/lib/jquery.js"></script>
```

Note Refer to the sample XSL file in the XSLT zip folder (`demoxslt.xml`).

Note If there is a single CSS file, then name the CSS file the same as the base script file and it will be automatically picked up. For example, if the base filename is `main.js` then name the CSS filename as `main.css`. In this case the CSS file should be present in the same directory as that of the base script file.

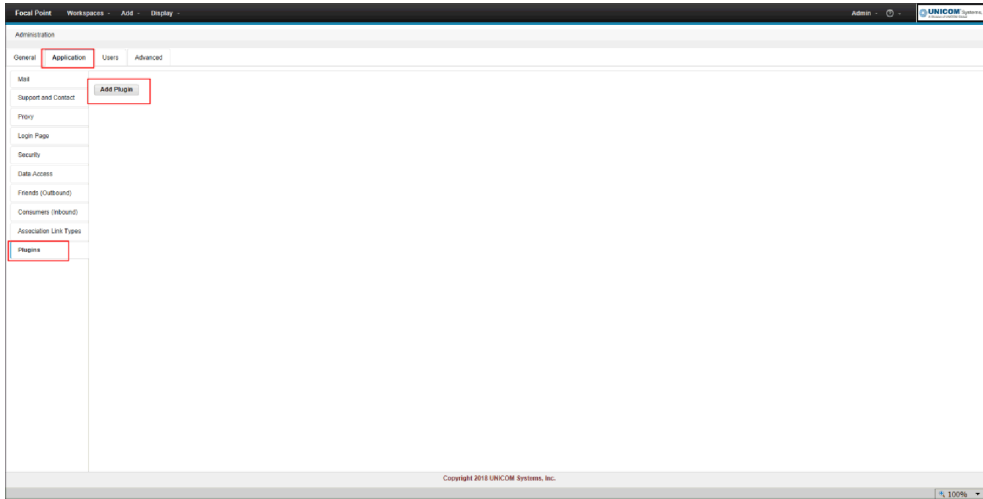
Step 4 Create the Plugin

The final step is to create a zip file of the directory `MYDIR` and upload it as a plugin.

Steps to upload a Custom Chart Plugin

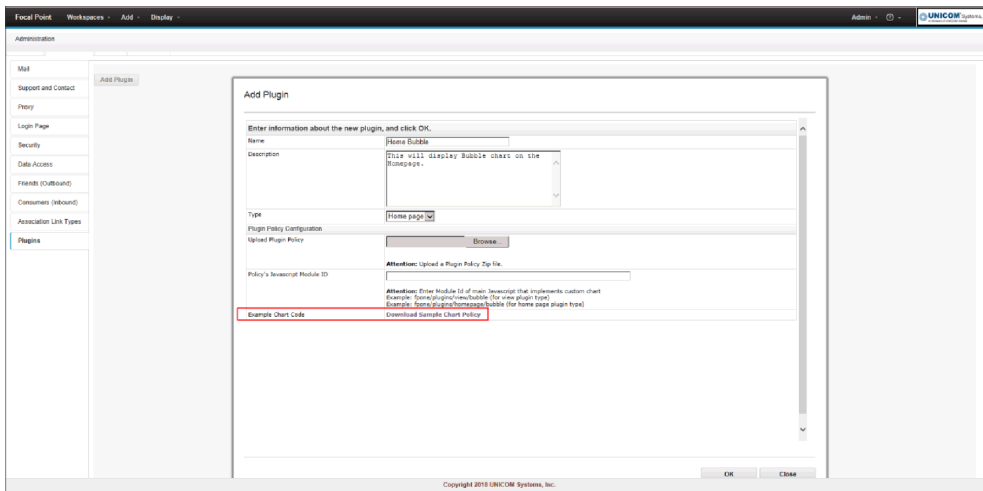
Step 1 Go to the Configure Plugin page

Login as Admin and go to **Administration > Application Tab > Plugins** and then click **Add Plugin**.



Step 2 Download the sample Chart Policy

Once the Plugin form is displayed, click on the link **Download Sample Chart Policy**.



Step 3 Unzip the file SampleChartPolicy.zip

Unzip the file `SampleChartPolicy.zip` to display the contents.

Name ^	Date modified	Type	Size	
JavaScriptAPIDoc	27-04-2018 18:36	File folder		
Bubble1.zip	02-02-2018 15:17	KuaiZip ZIP Archive ...	74 KB	
Bubble2.zip	02-02-2018 15:18	KuaiZip ZIP Archive ...	376 KB	
CustomPluginChartUsageGuide.txt	13-03-2018 17:18	Text Document	9 KB	
Rational_Focal_Point_RESTful_API_Referen...	27-04-2018 18:08	PDF File	204 KB	
readme.txt	27-04-2018 18:04	Text Document	4 KB	
XSLT_DEMO_WORKSPACE .fpz	27-03-2018 21:19	FPZ File	1,053 KB	
XsltTransform.zip	26-04-2018 18:57	KuaiZip ZIP Archive ...	17 KB	

Step 4 Import the sample workspace (optional step)

Note This step is optional and is needed only if you want to deploy the XSLTdemo plugin.

Login as Admin and import the sample workspace XSLT_DEMO_WORKSPACE . fpz file.

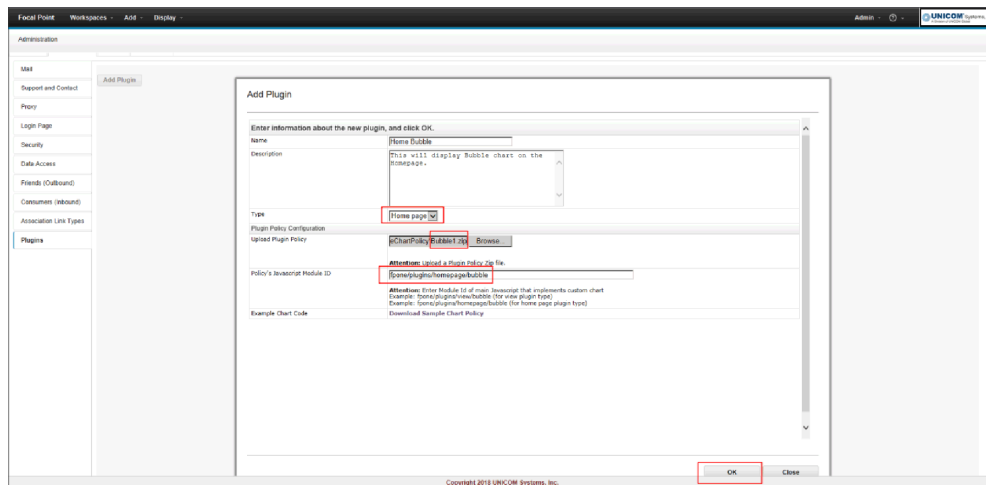
Step 5 Upload the Plugins

Now proceed to fill the **Add Plugin** Form with all the necessary plugin information as shown below.

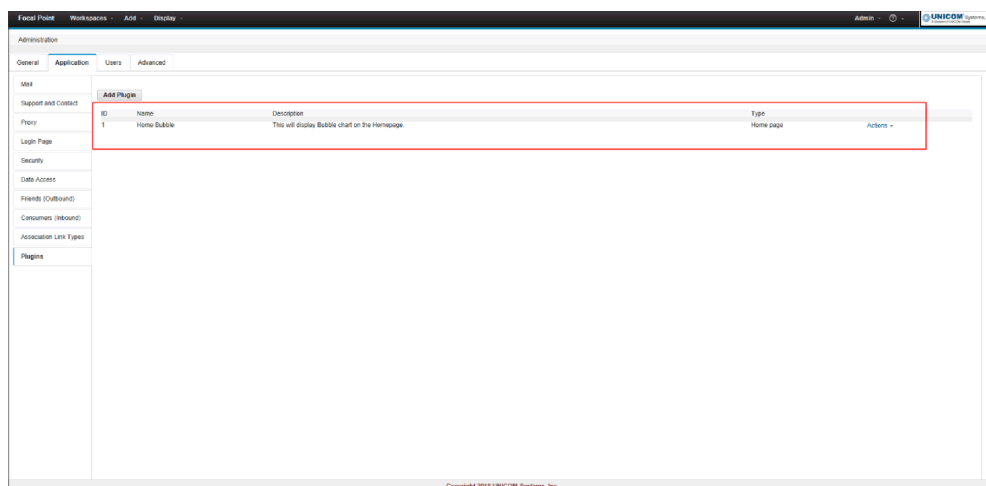
Upload Plugin 1

To display a bubble chart on the home page, select `Bubble1.zip` file against **Upload Plugin Policy**.

Provide 'Policy's Javascript Module ID' as `fpone/plugins/homepage/bubble`. Click on the **OK** button once the setup is complete.



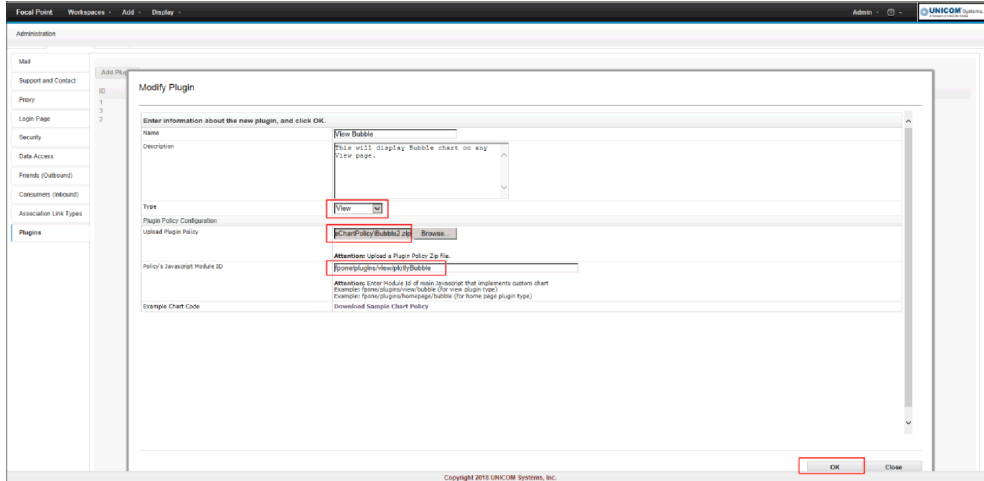
Once the upload process is successful, the "Plugins" page displays an entry as shown in the following screenshot:



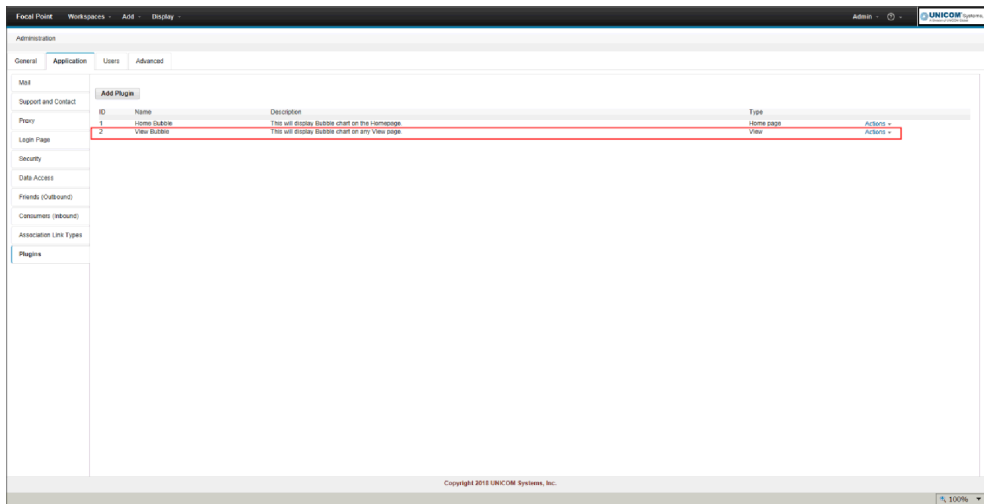
Upload Plugin 2 (optional step)

To display a bubble chart on a view page, select `Bubble2.zip` file against 'Upload Plugin Policy'. Entry against 'Policy's Javascript Module ID' will be `fpone/plugins/view/plotlyBubble`.

Click on the **OK** button, once the setup is complete.



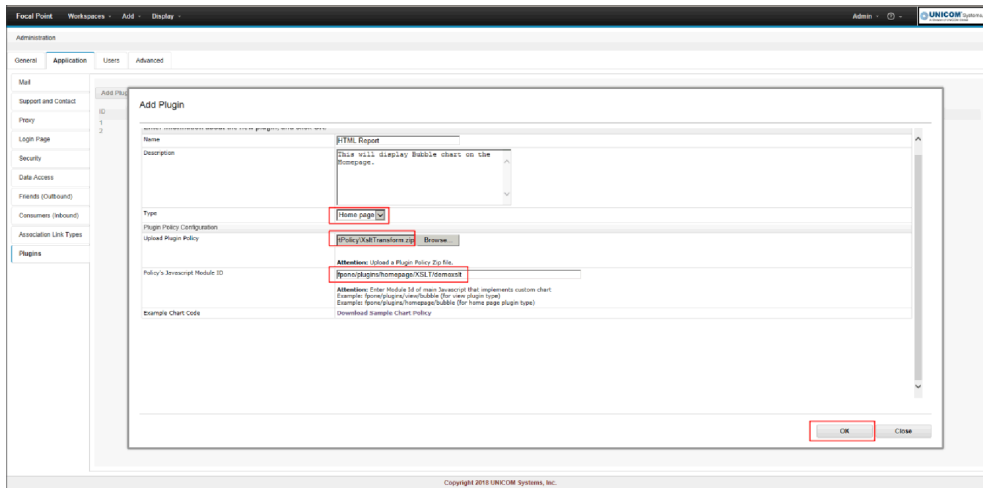
Once the upload process is successful, the "Plugins" page displays an entry as shown in the following screenshot:



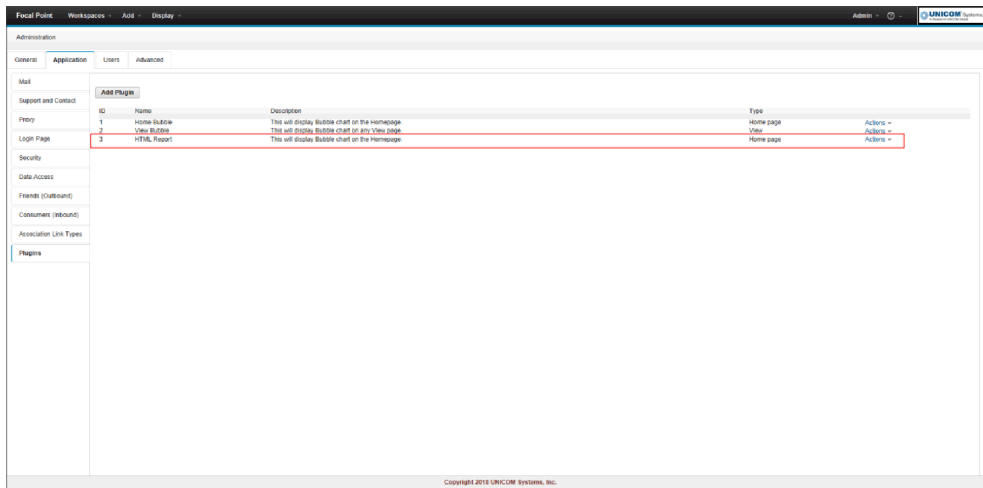
Upload Plugin 3 (optional step)

For displaying a HTML report on the home Page, select `XslTransform.zip` file against **Upload Plugin Policy**. Provide the value `fpone/plugins/homepage/XSLT/demoxslt` against 'Policy's Javascript Module ID'.

Click on the **OK** button, once the setup is complete.



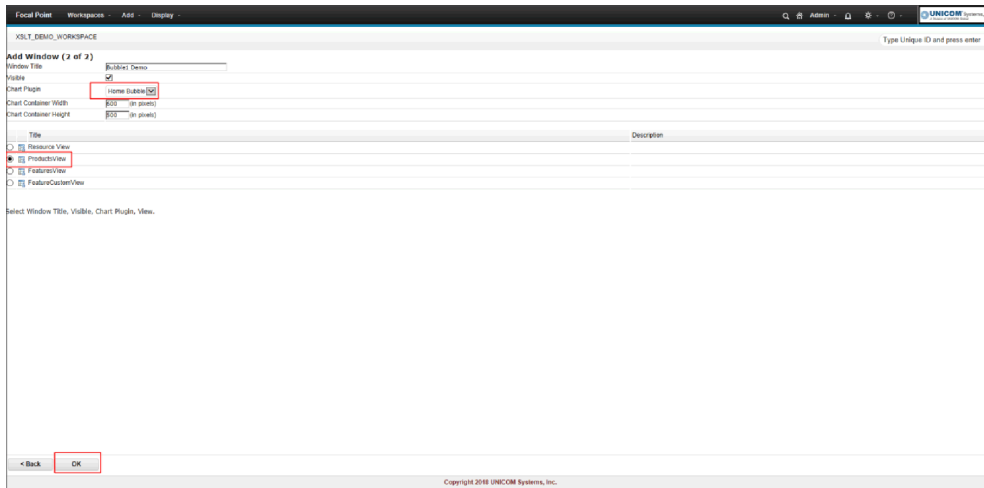
Once the upload process is successful, the “Plugins” page displays an entry as shown in the following screenshot:



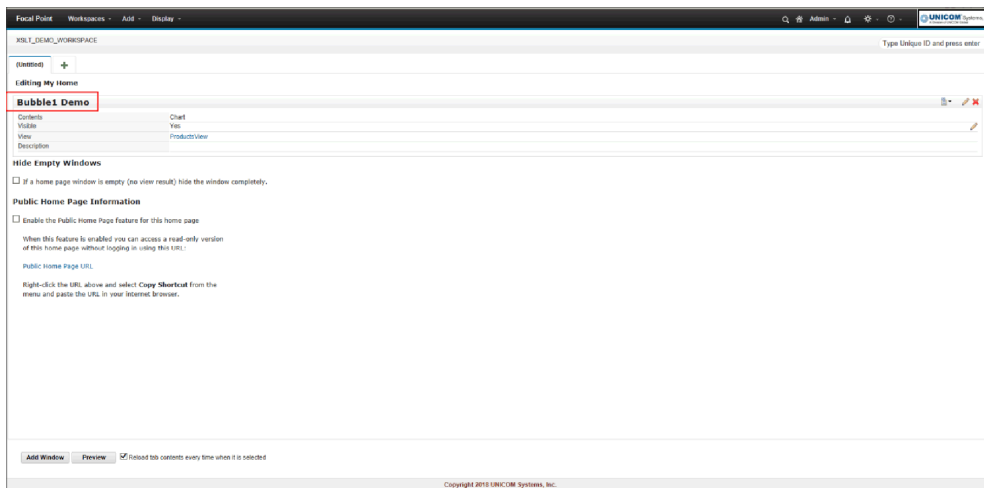
Step 6 Configure the plugin on the home page

Open **XSLT DEMO WORKSPACE**, go to **Homepage Settings > Add window** and click on the **Chart** option. The Add window form will be displayed.

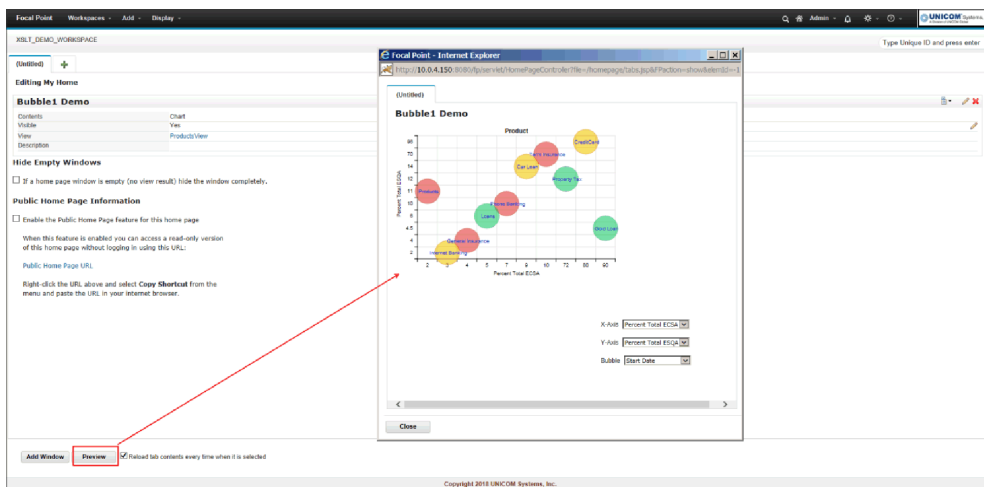
Proceed by filling in the necessary details like Window title, Chart Plugin and View.



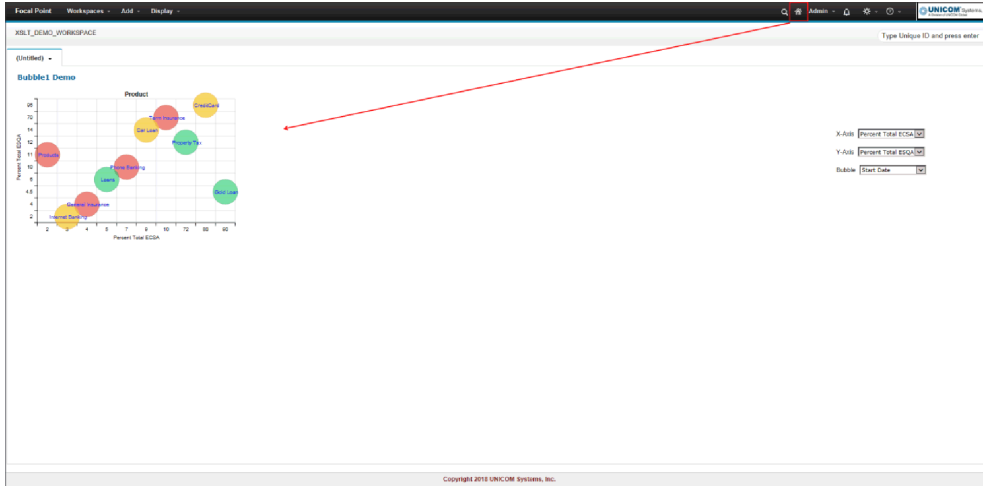
On clicking the **OK** button, the below page is displayed.



Click on the **Preview** button to see the chart being displayed.



Click on the Home icon to see the home page.



Change the values on X-axis, Y-Axis and bubble to see the changes being applied to the chart.

Step 7 Configure the plugin on the Homepage for XSLT plugin

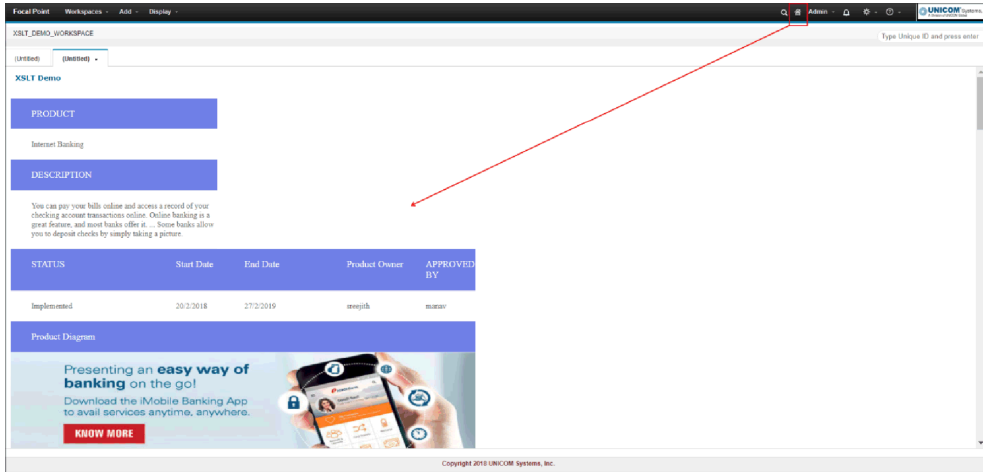
Open the **XSLT DEMO WORKSPACE**, go to **Homepage Settings > Add window** and click on the **Chart** option. The Add window form will be displayed.

Proceed by filling in the necessary details like Window title, Chart Plugin, and View. Set the height=5000 and width=900.

Click on the **OK** button.

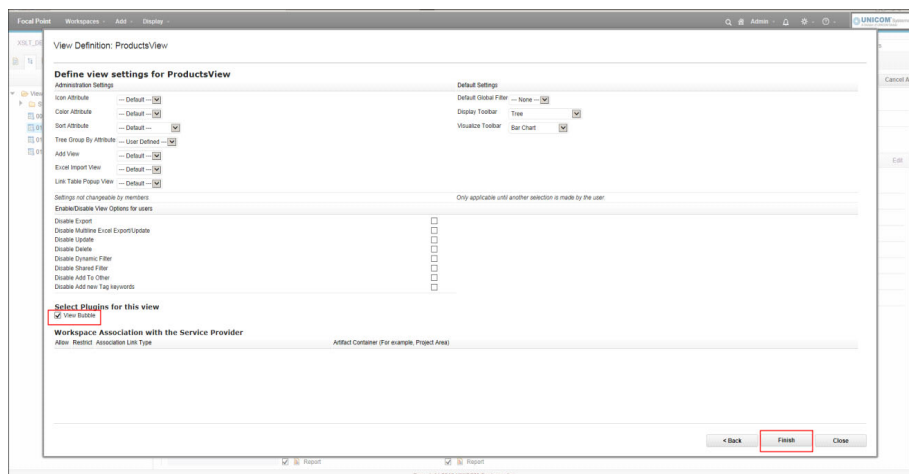
Click on the **Preview** button to see the chart being displayed.

Click on the Home icon to see the Home Page.

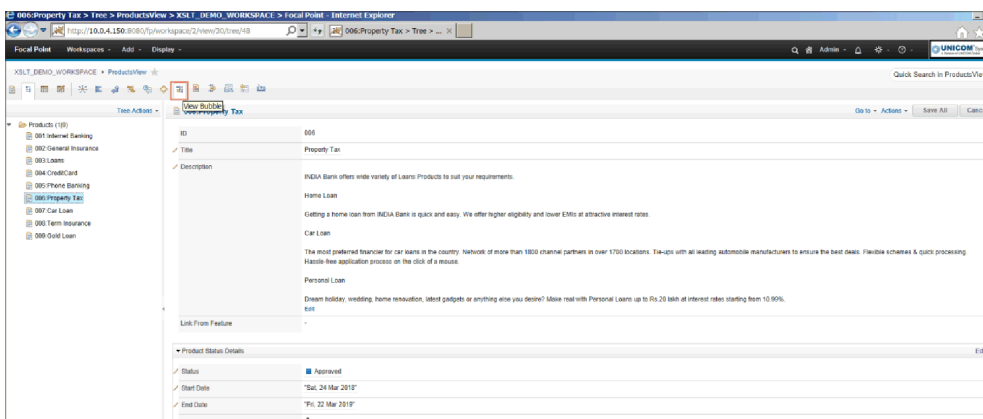


Step 8 Configure the plugin on a view page

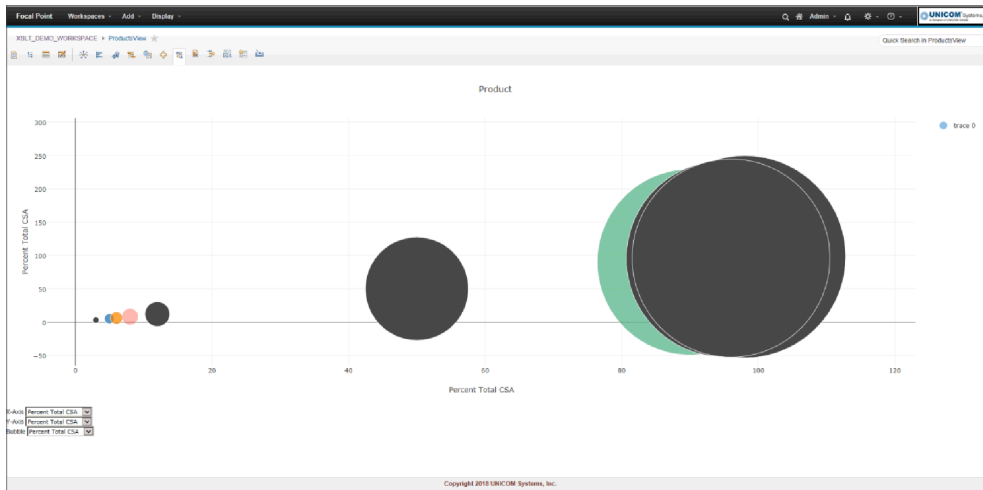
Go to Configure views, select any view and Edit the view definition and click **Next** twice to reach the Define view settings page. Select **Plugins for this view** by selecting the checkboxes for appropriate plugin and click **Finish** as shown below.



Now go to Display View and select the view, and click on the icon as show below to display the chart on the view Page.



On click, the following chart is displayed.



CUSDATA/DAO object

This section provides information about the values present in CUSDATA/DAO object. Some of the useful information in CUSDATA object is given below.

DAO object in custom script can be accessed via **JSON Object > CUSDATA.dao**.

```
{
  "dao": {
    "activeFilterId": -1,
    "moduleNamePlur": "Products",
    "viewId": 30,
    "fpresourceUrl": "http://10.0.4.150:8080/fp/resources",
    "ncModuleNamePlur": "Products",
    "moduleName": "Product",
    "treeElement": 48,
    "attrsInfo": [],
    "ncModuleName": "Product",
    "moduleId": 13,
    "userId": "13",
    "workspaceId": 2,
    "divId": "fp-maincontainer"
  },
  "basicFpDataUrl": "http://10.0.4.150:8080/fp/resources/workspaces/2/modules/13/elements/?view=30&includeAttributes=true&optimize=true",
  "namespaceFPUrl": "http://10.0.4.150:8080/fp/namespace/workspaces/2/modules/13/views/30/elements",
  "attrNames": [],
  "attrTypes": [],
  "attrRealNames": [],
  "attrIds": [],
  "pluginBasePath": "../../../one/js/modules/plugins/view/"
}
```

Note The Attribute Info array contains array length() equal to the number of attributes in the module.

CUSDATA API Information

Open the `JavaScriptAPIDoc` folder unzipped from `SampleChartPolicy.zip`. Open the `index.html` File in Browser and all methods could be browsed in detail. Some of the common methods are given below.

```
getAttribIdFromName (attribName) > {String}
```

Returns the attribute Id for a given Attribute Name.

Parameters:

Name	Type	Description
<code>attribName</code>	String	Attribute name.

Returns:

`attribId` - Attribute ID

Type: String

```
getAttribNameFromId (attribId) > {String}
```

Returns the attribute Name for a given Attribute Id.

Parameters:

Name	Type	Description
<code>attribId</code>	String	Attribute ID.

Returns:

`attribName` - Attribute Name

Type: String

```
getAttrNamesFromDAO (types) > {Array}
```

Finds the attribute names based upon the types specified. If no types are specified, then all 'integer' and 'float' attributes are retrieved.

Parameters:

Name	Type	Description
Types	Array	Array of type of attributes specified to prepare the attribute list.

Returns:

keys - Array of names of attributes

Type: Array

```
getBasicFpDataUrl() > {String}
```

Creates basic URL.

Returns:

Return basic Focal Point® DATA URL

Type: String

```
getCurrentPluginId() > {String}
```

Returns the plugin id from DAO object.

Returns:

Plugin id

Type: String

```
getCustomFpDataUrl(ATTR, conditions) > {String}
```

Generates a Focal Point® REST URL to retrieve data for the attributes from the view in which the plugin is operating. Users can provide a list of attributes for which data needs to be retrieved. It is also possible to provide attribute conditions so that the elements retrieved from the REST URL are filtered based on these conditions.

Refer to the Focal Point® REST API for details on how attribute conditions can be specified. This method returns a Focal Point® REST URL after the list of attributes and conditions are added to the base URL obtained through the call `getBasicFpDataUrl()`.

Parameters:

Name	Type	Description
ATTR	Array	Array containing the names of the attributes.
conditions	Array	Array containing the attribute conditions.

Returns:

Returns custom URL

Type: String

```
getData(ATTR, conditions, callback) > {Void}
```

This method returns the collection of attribute values retrieved from the current view by making a REST call to the URL obtained from the method `getCustomFpDataUrl()`. Users can provide a list of attributes for which data needs to be retrieved.

Note Only Integer, Float, Date and Text attributes are supported here.

It is also possible to provide attribute conditions so that the elements retrieved from the REST URL are filtered based on these conditions.

Refer to the Focal Point® REST API for details on how attribute conditions can be specified. If the user is looking for the unfiltered response, or is looking for information of attributes other than Integer, Float, Date or Text, then it is recommended to use the method `getFPRestData` which will return the entire response in JSON format.

Parameters:

Name	Type	Description
ATTR	Array	Array containing the names of the attributes.
conditions	Array	Array containing the attribute conditions.
callback	Function	Callback function with chart data.

Returns:

Type: Void

```
getFPRestData(URL, callback, outputFormat) > {Void}
```

This method returns data in JSON Object or XML format from any Focal Point® REST URL.

Note By default the response would be in JSON Format.

User can provide a REST URL as a parameter. To retrieve element and attribute information from the current view in which the plugin is operating, users can pass `@param{basicFpDataUrl}` as defined above. If data needs to be retrieved from any other URL, then users needs to construct the URL accordingly.

Refer to the Focal Point® REST API documentation in the help section for details on how a GET URL can be constructed.

Parameters:

Name	Type	Description
URL	String	Focal Point® REST URL to fetch DATA.
callback	Function	Callback function with Focal Point® REST response data.
outputFormat (optional)	String	Value should be either "json" or "xml". This parameter is optional. If omitted, output would be in JSON Object format.

Returns:

Type: Void

`getPref() > {Object}`

Retrieves user preference data from window object.

Returns:

User preference data as a JSON object.

Type: Object

`getPrefData(callback) > {Void}`

Retrieves the user preference data from the Focal Point® Database.

Parameters:

Name	Type	Description
callback	Function	Callback functions with user preference data as a parameter.

Returns:

Type: Void

`postPrefData(callback) > {Void}`

Saves the user preference data into the Focal Point® Database using a HTTP POST call. The data to be saved will be retrieved from the window object using the `getPref()` call. Users should have called `savePref()` to store user preference to the window object before calling this method.

Parameters:

Name	Type	Description
callback (optional)	Function	Callback function to handle the success message from the post call. This parameter is optional.

Returns:

Type: Void

```
resize() > {Void}
```

Sets the window dimensions for view and home page.

Returns:

Type: Void

```
savePref(JsonSettingsData) > {Void}
```

Saves the user preference data into window.fp object for further use.

Parameters:

Name	Type	Description
JsonSettingsData	Object	User preference data in JSON format.

Returns:

Type: Void

```
transform(xmlData, userDefinedXslFilePath) > {Void}
```

Applies the XSL transformation on the XML data for view or home page.

Parameters:

Name	Type	Description
xmlData	Object	Data in XML format.
userDefinedXslFilePath	String	Relative XSL File path defined by the user.

Returns:

Type: Void

Data types

This is a description of the data types used by the operations of the Web Service API. They are defined in the WSDL using an inline XSD Schema. When using an object-oriented programming language that can generate code from a WSDL document (for example Microsoft.NET or the Java Apache Axis framework), these data types are typically transferred into classes.

ID

An ID is the combination of a localId and a workspaceId. Focal Point® is divided into several workspaces, each with its own identifier. Objects (views, elements, attributes etc) in each workspace have their own identifier, but it is only unique within that workspace. In order to uniquely identify an object you need both its own identifier (i.e the localId) and the identifier of the workspace (i.e the workspaceId) it is part of. The data type ID is these two identifiers paired together. For example there can be an element with localId 200 in workspace 2 and another element with localId 200 in workspace 3.

Workspace

A Workspace contains information about a workspace in Focal Point®. It has an identifier (and integer), a title (the name of the workspace) and a description.

View

A View contains information about a view in Focal Point®. It has an ID, a title (the name of the view) and a description.

Element

The data type Element is used to represent an element in Focal Point®. It consists of an ID (that uniquely identifies the element) and a list of Attributes.

Attribute

An Attribute is used to maintain the attribute values for an Element. It consists of an "attributeSetupId" which is an ID that references the AttributeSetup that defines the attribute, a "value" which is an AttributeValue that contains the actual value of the attribute and a "displayValue" which is a simple string representation (if possible) of the attribute value.

The displayValue property should be regarded as a read-only property. When updating or creating an element the displayValue of an Attribute will be ignored, it is only the AttributeValue that is used.

If the value of an attribute is empty/null both the AttributeValue property and displayValue will be null.

AttributeValue

AttributeValue is in itself an empty data type. Instead there are a number of data types that extend AttributeValue. In an object-oriented environment this translates to a superclass/subclass structure.

CheckBoxValue

The CheckBoxValue has a Boolean property called selected that tells if the CheckBox attribute is selected or not.

ChoiceValue

The ChoiceValue has an integer that identifies the selected choice. The full list of choices and their names is retrieved from the Attribute's AttributeSetup.

DateValue

The DateValue has an xsd:dateTime that represents the value of the date attribute. For most date attributes in Focal Point® only the date and not the time point is stored, but in the DateValue the hour and minute is included as well. In cases where they haven't been set in the Focal Point® server, the time point 00:00 will be returned from the Web Services.

The time point part will be ignored when setting or updating an attribute value.

FloatValue

The FloatValue has an xsd:double that represents its value.

IntegerValue

The IntegerValue has an xsd:long that represents its value.

LinkValue

The LinkValue is represented by an ID that identifies another element in Focal Point®.

ListValue

The ListValue does not contain any values itself. Instead it has a list of values. It can either be a list of LinkValues or a list of TextValues

MultiChoiceValue

The MultiChoiceValue has a list of integers that represents the selected choices. The full list of choices and their names is retrieved from the Attribute's AttributeSetup.

TextValue

The TextValue contains a string with the value of the text attribute. It will be equal to displayValue.

Any formatting used will be ignored.

UniqueIdValue

The UniqueIdValue is represented by a string.

UrlValue

The UrlValue is represented by a string.

VersionValue

The VersionValue is represented by a string.

FileValue

The FileValue contains a list of FileDataValues. The FileDataValue contains the name, content type and length of a file. Each FileDataValue also has a file number which is an identifier for that particular file within the file attribute. Note that the actual file content is not included when reading a file attribute (for example when using GetElement or GetElements). Retrieving the file content can be done using GetFileContent. When writing a file (when using AddElement or UpdateElement) the file content shall be included. If the supplied file number is found in the existing file attribute (on the server) the file will be replaced with the supplied data, if the file number is not found a new file will be added.

AttributeSetup

The AttributeSetup is the definition of an attribute. An AttributeSetup is always referred to by an Attribute. It contains an ID, a "title" (the name of the attribute), a "description", a "type" (an AttributeType) and a "mandatoryName".

The mandatoryName is a string that is only used for attributes that are mandatory (for example title, last changed by). Since it is possible to change the name for any attribute in Focal Point®, mandatoryName is a name that does not change. For example the mandatoryName of the description attribute will always be "Description" even if the title has been changed to something else.

Another way to determine which attribute is the title, description, or prefix attribute is reading the AttributeSetupInfo in a ViewInfo or ElementSet.

ElementSet

An ElementSet is a container of Elements and some information about the view they were retrieved from. It has a list of Elements, a list of AttributeSetups reflecting the attribute setups for the view that the elements were retrieved from and an AttributeSetupInfo.

AttributeType

The AttributeType data type is used by AttributeSetup to tell which type attribute values based on that attribute setup will be. AttributeSetup is an enumeration and can be any of the following values: Text, Link, Date, CheckBox, List, Choice, Float, Integer, MultiChoice, Url, UniqueId, or Version.

Example: If an AttributeSetup has the AttributeType Date any Attribute that refers to the AttributeSetup will have a DateValue value.

AttributeSetupInfo

The AttributeSetupInfo data type contains information about a set of AttributeSetup retrieved from a view. It contains three IDs that each point to an AttributeSetup. This is used to tell which attribute is the title, description, and prefix attribute. It is also possible to determine this by reading the mandatoryName property of an AttributeSetup (see the description of AttributeSetup).

ChoiceSetup

The ChoiceSetup data type is an extension of the AttributeSetup data type. It includes everything an AttributeSetup has got and adds a list of ChoiceSetupItems. The ChoiceSetup is used when the AttributeSetup's AttributeType is Choice. The ChoiceSetupItems contain information about all available choices for a choice attribute.

ChoiceSetupItem

The ChoiceSetupItem data type contains information about a choice for a choice attribute. It has an integer which is an identifier for this choice and a "title". The id is used to match ChoiceSetupItems and ChoiceValues.

ViewInfo

The ViewInfo data type contains a list of AttributeSetups and an AttributeInfo. That can be useful if you want to know which attributes that are visible in a view without first retrieving an ElementSet, for example if you want to create a new Element.

HistoryEntry

A HistoryEntry describes a previous change of an attribute. It consists of four parts: “username” the full name of the user who made the change, “date” the date of the change, “elementId” the ID of the element that was changed and “attribute” the changed attribute with the value it had at the time of the change.



www.unicomsi.com

We welcome feedback on our documentation. Please email us at:
tech.authors@unicomsi.com

www.unicomglobal.com