

CA PPM v13.3  
CA PPM v14.1

Hierarchical Views r2  
Administration Guide

**Table of Contents**

[1. Version History 3](#_Toc415052568)

[2. Introduction 4](#_Toc415052569)

[3. Deploying a new Hierarchical View 7](#_Toc415052570)

[3.1. Create a new Hierarchical View Record 7](#_Toc415052571)

[3.1.1. How to create your Query 12](#_Toc415052572)

[3.1.2. Tree Heat Maps 13](#_Toc415052573)

[3.1.3. Org Charts 13](#_Toc415052574)

[3.1.4. Word Trees 13](#_Toc415052575)

[3.1.5. Sankey Diagrams 14](#_Toc415052576)

[3.2. Create a new HTML Portlet 15](#_Toc415052577)

[3.3. Create a custom Tab on the Page Layout 17](#_Toc415052578)

[4. Known Issues and Limitations 19](#_Toc415052579)

# Version History

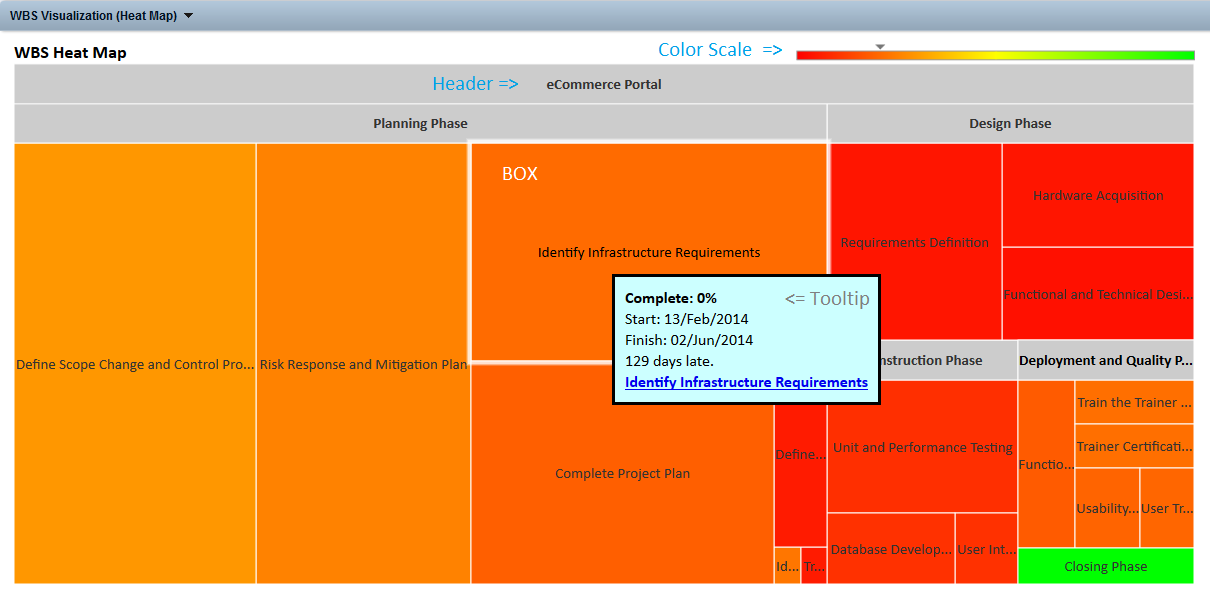
|  |  |  |  |
| --- | --- | --- | --- |
| Author | Date | Version | Comments |
| Alexandre Assis | 09/Jan/2015 | 1.0 | Initial Release |
| Alexandre Assis | 25/Mar/2015 | 2.0 | Second Release, added Word Trees and Sankey Diagrams |

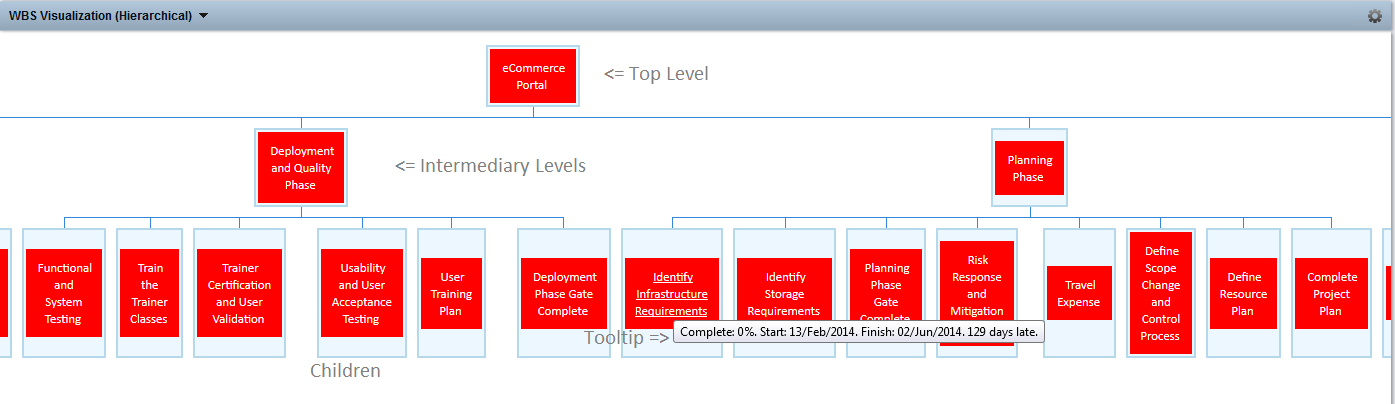
# Introduction

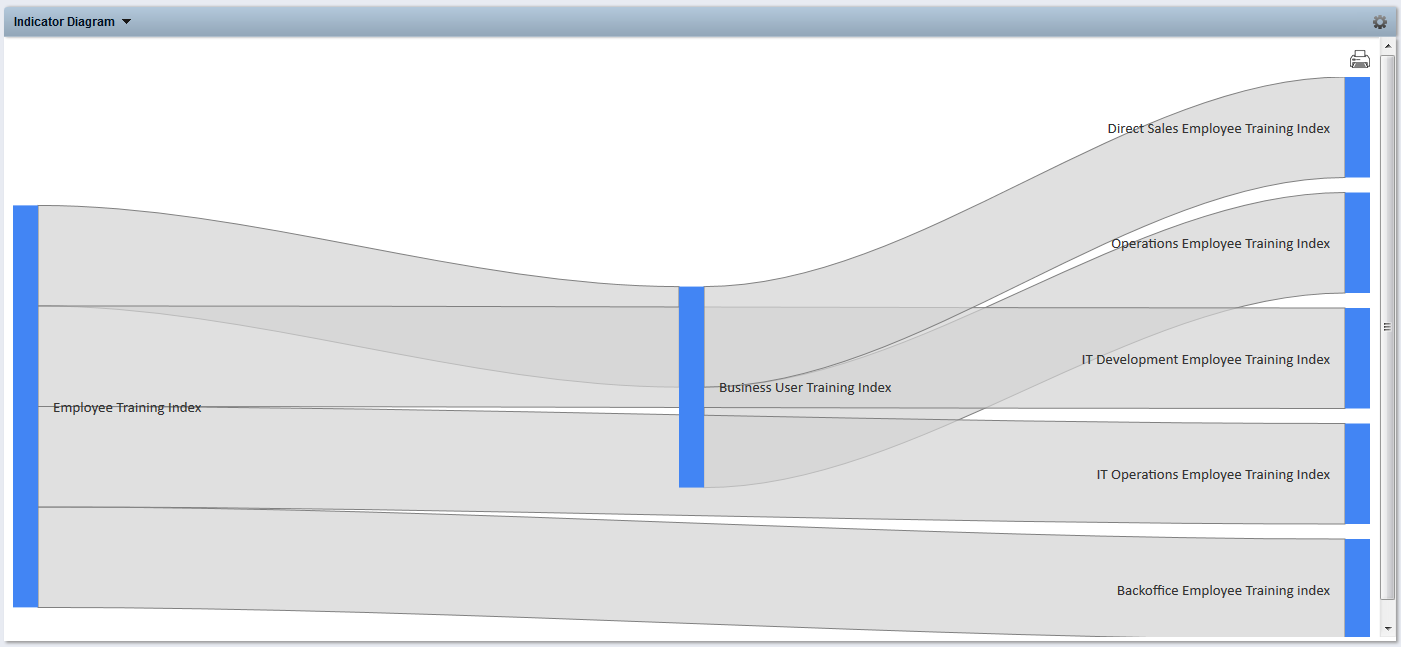
The Hierarchical Views package allows you to create custom hierarchical views and deploy them on Clarity Pages using HTML Portlets.

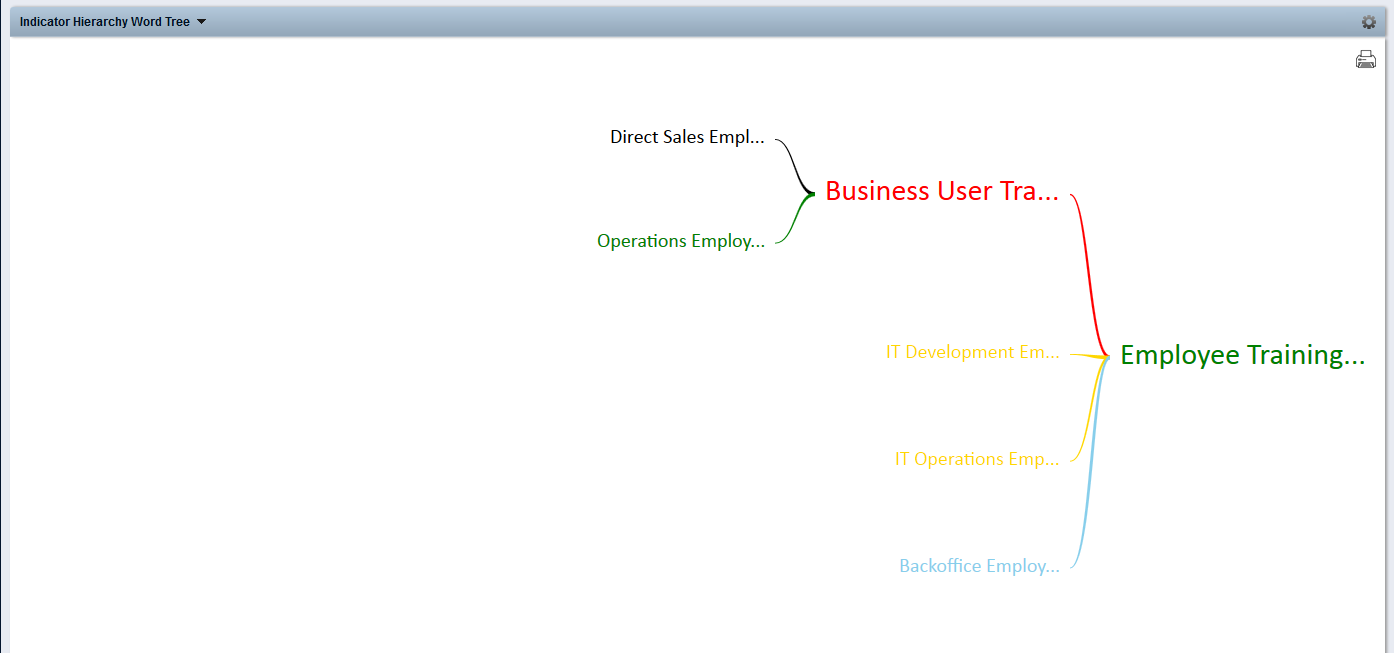
On this Version two types of Hierarchical Views are supported: the Tree Map (also known as Tree Heat Map) and the Organizational Chart.

See examples below.









# Deploying a new Hierarchical View

To create a new Hierarchical View, follow these steps.

## Create a new Hierarchical View Record

Login: Admin

Start in Home -> Custom Objects -> Hierarchical View List



|  |  |
| --- | --- |
|  | Click New Hierarchical View |
|  | Enter a new Name, a new ID and choose the Hierarchy Type.  Standard Naming for the ID:  hiev\_<package>XXXXXX\_<type>  hiev: is the Hierarchical Views package identifier  package: is the code used to identify your custom package (i.e. strat for Strategic Planning)  XXXXXX: is the portlet identifier (i.e. what is thie view about)  type: **hm** for Heat Maps, **org** for Org Charts, **wt** for Word Trees and **sd** for Sankey Diagrams    Click SAVE |
|  | On the Properties Page, enter the **Required Information.**  Thissection contains attributes that will jointly form an SQL text that will be used to bring data to your portlet.  You have up to 4 levels available, and you must follow the provided templates.  You need at least one of the SQL Parts filled.  More information on how to build your query will be available later on this topic.  Click the appropriate TAB depending on the type of Chart you are creating. |
|  | **Heat Map Options**  On the **Basic** section you will find the most frequently changed properties to guide the behavior of a Tree Heat Map.  On the **Advanced** section you will find additional Formatting options.  To get more information on each option visit [Google Charts Tree Map](https://developers.google.com/chart/interactive/docs/gallery/treemap).    The most important:  Max Depth: maximum levels you want on the same view, without the need to drill down.  Min, Mid, Max Color: define the Color Scale. Min Color is the color assigned to the lowest values, Max Color is the color assigned to the highest values. Mid Color is the one in the middle. These three colors will result in a color scale.  Select “Drill to New Page” to open a new Browser Tab when Drilling Down from your chart.  All attributes have standard values that can later on be adjusted if required  Click SAVE AND RETURN |
|  | **Org Chart Options**  Choose the Chart Node Size (small, medium or large).  Select “Drill to New Page” to open a new Browser Tab when Drilling Down from your chart.  All attributes have standard values that can later on be adjusted if required  Click SAVE AND RETURN |
|  | **Word Tree Options**  Choose the Font Family and Maximum Font Size.  Choose the Word Tree Type –  Effect->Cause or Suffix trees draw from right to left representing information that is aggregated from the detailed left nodes to the right main node.  Cause -> Effect or Prefix trees draw from left to right representing information that is aggregated from the detailed right nodes to the left main node.  All attributes have standard values that can later on be adjusted if required  Click SAVE AND RETURN |
|  | **Sankey Diagram Options**  On the **Basic** section you will find the most frequently changed properties to guide the behavior of a Sankey Diagram.  On the **Advanced** section you will find additional Formatting and Behavior options.  To get more information on each option visit [Google Charts Sankey](https://developers.google.com/chart/interactive/docs/gallery/sankey) Diagram. |

### How to create your Query

The query is configured on the Hierarchical View record.

The final query will always have this format:

Select

X.NodeName, X.NodeID, X.NodeParentName, X.NodeParentID, X.NodeLevel, X.NodeSize, X.NodeColor,

X.NodeColorName, X.NodeLinkURL, X.NodeAdditionalInfo

from (

… Your First Query …

Union

… Your Second Query …

Union

… Your Third Query …

Union

… Your Forth Query …

) X ORDER BY X.NodeLevel

At least One query is required and you have a Maximum of Four queries. Typically you will use a query for each level.

Each one of your queries needs to have this format:

Select

TABLE.<NAME ATTRIBUTE> NodeName,

TABLE.<INTERNAL ID ATTRIBUTE> NodeID,

TABLE.ATTRIBUTE NodeParentName,

TABLE.ATTRIBUTE NodeParentID,

TABLE.ATTRIBUTE NodeLevel,

TABLE.ATTRIBUTE NodeSize,

TABLE.ATTRIBUTE NodeColor,

TABLE.ATTRIBUTE NodeColorName,

TABLE.ATTRIBUTE NodeLinkURL,

TABLE.ATTRIBUTE NodeAdditionalInfo

From

TABLE

Where

TABLE.<INTERNAL ID ATTRIBUTE>= %internal\_id%

%internal\_id% will be replaced in Runtime with the InstanceID of the current page.

There are specific information for each Hierarchy Type.

### Tree Heat Maps

When you use Tree Maps, pay attention to the following when building your query:

**NodeParentName** must be **null** for the parent node (starting “Node”)

**NodeParentID** is ignored for Tree Heat Maps

**NodeSize** is a number that determines the relative size of the Box in the Tree Heat Map.

**NodeColor** is a number that Tree Map will use to determine the color of the Box using the Color Scale.

**NodeColorName** is ignored for Tree Maps.

**NodeLinkURL** is a string with a CA PPM Link (starting in /niku/nu) that allows you to drill down to the corresponding Object Instance.

**NodeAdditionalInfo** is a String in HTML Format containing any additional information you may want to present as a “Tooltip”.

### Org Charts

When you use Org Charts, pay attention to the following when building your query:

**NodeParentName** must be **null** for the parent node (starting “Node”)

**NodeParentID** is ignored for Org Charts

**NodeSize** is ignored for Org Charts

**NodeColor** is ignored for Org Charts

**NodeColorName** is a string containing an HTML, HEX or RGB color for each box

**NodeLinkURL** is a string with a CA PPM Link (starting in /niku/nu) that allows you to drill down to the corresponding Object Instance.

**NodeAdditionalInfo** is a String containing any additional information you may want to present as a “Tooltip”.

(Do not use HTML Formatting for Org Charts).

### Word Trees

When you use Word Trees, pay attention to the following when building your query:

**NodeParentName** is ignored on Word Trees.

**NodeParentID** must be **-1** for the parent node (main “Word”)

**NodeSize** is a number that determines the relative size of the Words in the Word Tree.

**NodeColor** is ignored for Word Trees.

**NodeColorName** is a string containing an HTML, HEX or RGB color for the Words and Links.

**NodeLinkURL** is ignored for Word Trees.

**NodeAdditionalInfo** is ignored for Word Trees.

### Sankey Diagrams

When you use Sankey Diagrams, pay attention to the following when building your query:

**NodeParentName** must be **null** for the parent node (starting “Node”)

**NodeParentID** is ignored for Sankey Diagrams

**NodeSize** is a number that determines the relative size (weight) of the link between Nodes.

**NodeColor** is ignored for Sankey Diagrams.

**NodeColorName** is ignored for Sankey Diagrams.

**NodeLinkURL** is ignored for Sankey Diagrams.

**NodeAdditionalInfo** is ignored for Sankey Diagrams.

## Create a new HTML Portlet

Now, create a new HTML Portlet in CA PPM Studio.

To easily identify your Hierarchical View Portlets, consider using the following naming standard for the portlet ID:

<package>\_XXXXXX\_hiev\_<type>

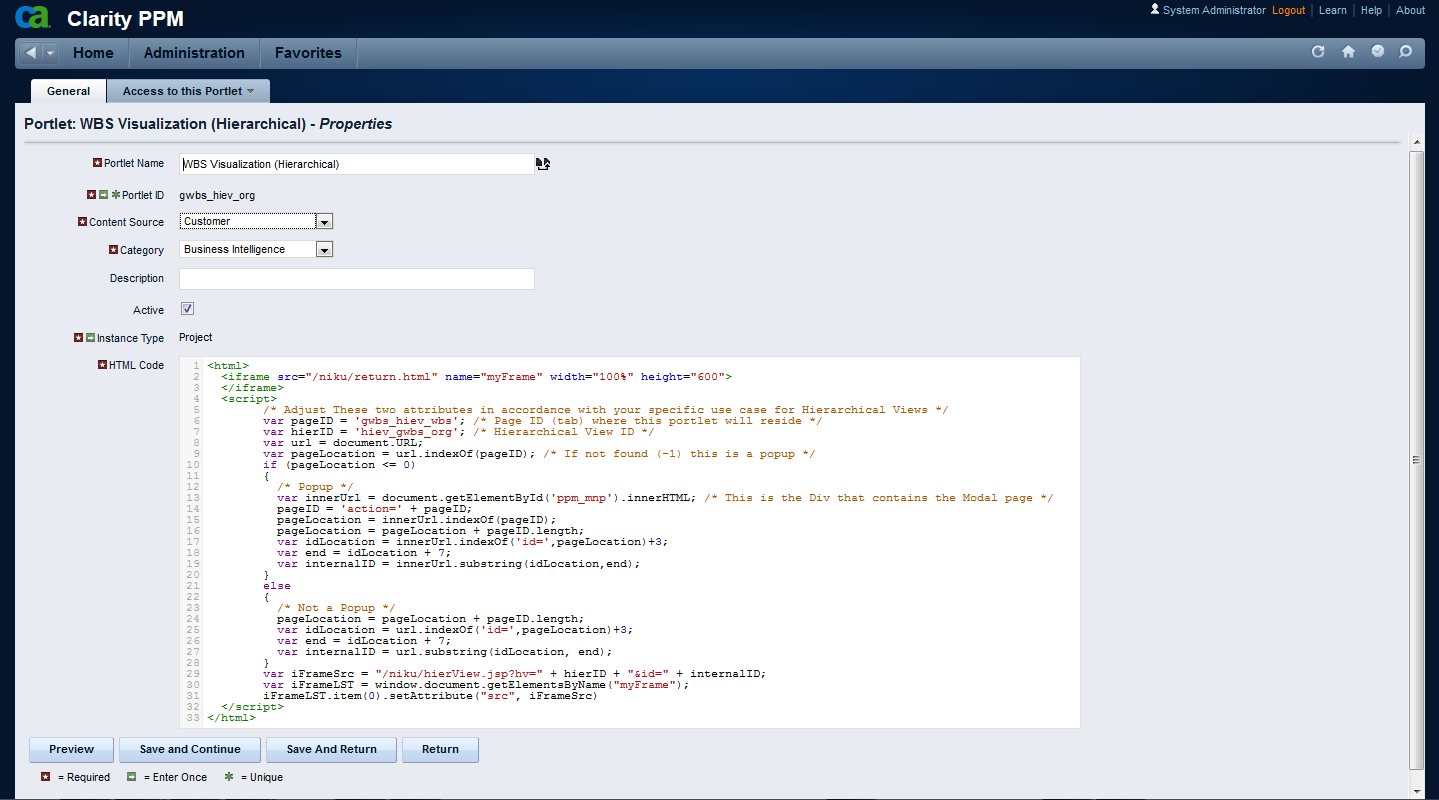
package: identifies your customization initials

XXXXXX: Portlet Identifier that tells us what this is about

hiev: a fixed-string that identifies this as a hierarchical view

type: is the Hierarchy Type: **hm** (Heat Map), **org** (Org Chart), **wt** (Word Tree), **sd** (Sankey Diagram)

Example: we have created a new portlet to show a graphical view of a wbs using the org chart. The Portlet id is **gwbs\_hiev\_org**



The HTML Portlet will always have the same text, which is listed below.

The only changes you need to do to this text are the following parameters, displayed in red on the HTML text:

**hierID** refers to the Hierarchical View record you created before.

**pageID** refers to the Tab that you will create for deploying this portlet.

<html>

<iframe src="/niku/return.html" name="myFrame" width="100%" height="600">

</iframe>

<script>

/\* Adjust These two attributes in accordance with your specific use case for Hierarchical Views \*/

var pageID = 'gwbs\_hiev\_wbs'; /\* Page ID (tab) where this portlet will reside \*/

var hierID = 'hiev\_gwbs\_org'; /\* Hierarchical View ID \*/

var url = document.URL;

var pageLocation = url.indexOf(pageID); /\* If not found (-1) this is a popup \*/

if (pageLocation <= 0)

{

/\* Popup \*/

var innerUrl = document.getElementById('ppm\_mnp').innerHTML; /\* This is the Div that contains the Modal page \*/

pageID = 'action=' + pageID;

pageLocation = innerUrl.indexOf(pageID);

pageLocation = pageLocation + pageID.length;

var idLocation = innerUrl.indexOf('id=',pageLocation)+3;

var end = idLocation + 7;

var internalID = innerUrl.substring(idLocation,end);

}

else

{

/\* Not a Popup \*/

pageLocation = pageLocation + pageID.length;

var idLocation = url.indexOf('id=',pageLocation)+3;

var end = idLocation + 7;

var internalID = url.substring(idLocation, end);

}

var iFrameSrc = "/niku/hierView.jsp?hv=" + hierID + "&id=" + internalID;

var iFrameLST = window.document.getElementsByName("myFrame");

iFrameLST.item(0).setAttribute("src", iFrameSrc)

</script>

</html>

## Create a custom Tab on the Page Layout

Open your Object Page Layout and create a new Tab.

To easily identify your Hierarchical View Pages, consider using the following naming standard for the page ID:

<package>\_hiev\_XXXXXX

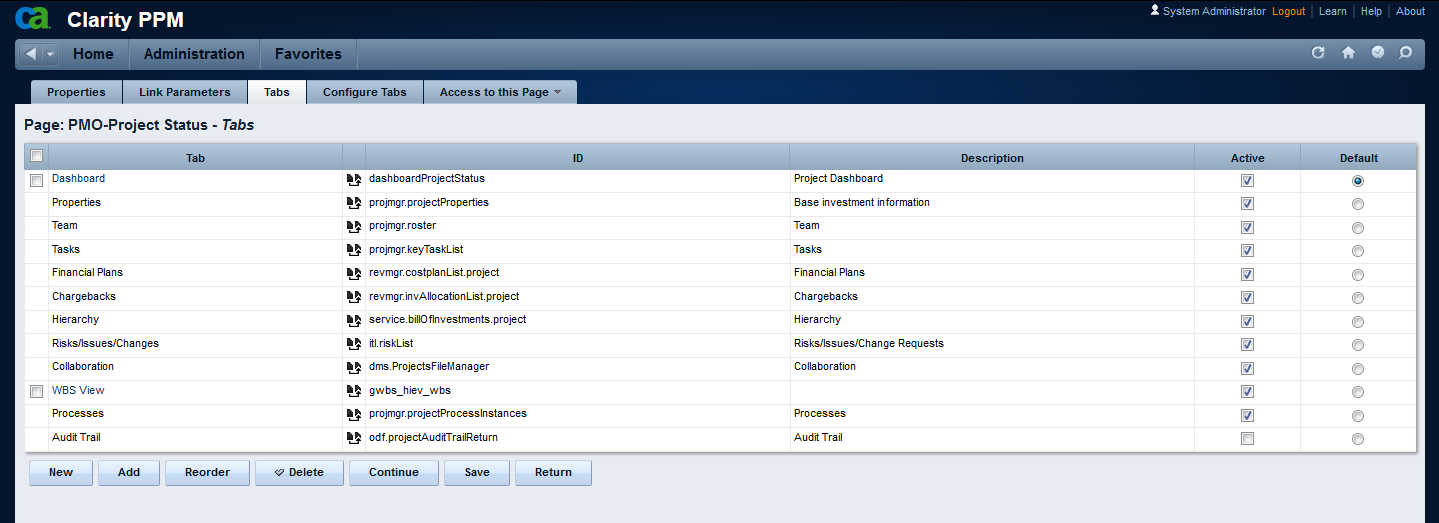
package: identifies your customization initials

hiev: a fixed-string that identifies this as a hierarchical view

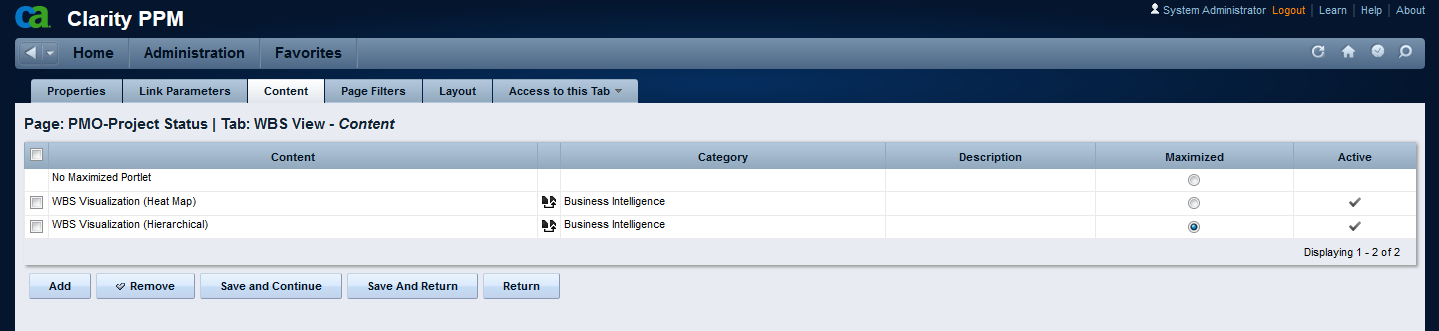
XXXXXX: Page Identifier that tells us what this is about

Example: On the **PMO-Project Status** we have created a new Tab called “**WBS View**” to deploy our new WBS Views, with the id

**gwbs\_hiev\_wbs**



Add your Hierarchical Views Portlets to this tab. Ideally, set one of the Portlets as Maximized.



# Known Issues and Limitations

1. You need access to the server to deploy this package as it requires a Custom JSP. On Demand customers may not get approval for this.
2. This package uses Google Charts. Therefore, you will need Internet access from your CA PPM server for them to work properly.
3. There is a maximum of four queries allowed for each Hierarchical View. That generally translates into four levels, but if you have a small enough query you could fit two levels in each, totaling eight levels. That would be the max.
4. Each HTML Portlet needs to identify the Page for which it was Built. That allows us to use CA PPM’s feature “Open as Popup”. Therefore, if a Portlet should be used in multiple pages you will need multiple portlets.
5. HTML Portlets are not Multi-Language. You need to hard-code one language in your query.