



# Working with Parameter Objects

---

Intellicus Enterprise Reporting and BI Platform



©Intellicus Technologies  
info@intellicus.com  
www.intellicus.com

---

Copyright © **2013** Intellicus Technologies

This document and its content is copyrighted material of Intellicus Technologies. The content may not be copied or derived from, through any means, in parts or in whole, without a prior written permission from Intellicus Technologies. All other product names are believed to be registered trademarks of the respective companies.

**Dated: - May 2013.**

## **Acknowledgements**

Intellicus acknowledges using of third-party libraries to extend support to the functionalities that they provide.

For details, visit: <http://www.intellicus.com/acknowledgements.htm>

---

## Contents

---

<b>Parameter objects .....</b>	<b>4</b>
To create a new parameter .....	4
Details on the dialog box .....	5
Multi Select Parameters .....	8
Setting Combo Source .....	9
Setting Search .....	12
Tree View .....	14
Linked Parameters .....	16
Validations .....	17
Modifying a Parameter Object .....	19
Deleting a Parameter Object .....	19
Parameter Objects from Data Model .....	19
Parameter Value Groups .....	21
To create group .....	23

# Parameter objects

The adhoc report gets its data by running pre-prepared query objects. If a query needs a value at run time, it may have a user (run time) parameter included in it. The values of Run time parameters are taken from the user while running the report.



**Note:** It is suggested that all required parameter objects are designed before proceeding to create a query object that may use a parameter object.

Parameters are stored on server and so can be used in one or more report and query objects.

To open Parameter page, click Navigation > Repository > Report Objects > Parameter.

The screenshot shows the 'Parameter' page in a software application. The page has a navigation bar at the top with 'Repository > Report Objects > Parameter'. Below the navigation bar, there are tabs for 'Query', 'Parameter', 'Parameter Value Groups', and 'Cube'. The 'Parameter' tab is active. The main area contains a form for creating or editing a parameter object. The 'Name' field is set to 'ParameterObject'. The 'ID' field is empty. The 'Prompt' field is empty. The 'Data Type' is set to 'Char' with a size of 30. The 'Format' field is empty. The 'Prompt Format' field is empty. The 'Default Value' field is empty. The 'Database Time Zone' and 'User Time Zone' fields are empty. The 'Description' field is empty. The 'Mandatory' checkbox is checked. The 'Visible' checkbox is checked. The 'Restrict to List' checkbox is checked. The 'Pass Values Using Tables' checkbox is checked. The 'Enable' checkbox is checked. The 'Forced' checkbox is checked. The 'Input Type' is set to 'Combo'. The 'Valid Characters' section is empty. The 'Invalid Characters' section is empty. The 'Add New' button is highlighted.

Figure 1: Parameter Page.

A blank parameter page opens.

Click **Add New** button to start working on a new parameter. Click **Open** to open a parameter.

## To create a new parameter

1. Click **Add New** button. All the entry boxes are cleared.
2. Specify values for the new parameter being set. Details are given below.
3. After providing all the required values, click **Save** button.

## Details on the dialog box

**Name:** A parameter has a unique Name, which will be used across the application.

A parameter also has a unique **ID** that application uniquely generates when parameter is saved.

**Note:** Name will be validated when you save the parameter.

**Prompt** is 'label' text of the parameter. It will be displayed on **Input Parameter Form (IPF)**, where it will help user understand what value should be entered for the parameter.

**Data Type:** This defines the data type of the parameter value. Select among:

- **Char:** Alphabets, numbers and special characters.
- **Number:** Digits and decimal point.
- **Date:** Date or part of date (like day, month, year).
- **Boolean:** A parameter that will accept one of the two pre-set values. Appears on screen as a checkbox.

**Size:** Number of digits or characters that user should be allowed to enter. (only for Char and Number data type).

**Format:** Select the format in which user should provide value for this parameter. Refer to online help to know more about Data Formats.

**Prompt Format:** Select the format in which user should provide value for this parameter at runtime.

For Date format, you can choose **Append Time** when **Apply Locale Default** is selected.

**Default Value:** Specify a value that is most likely to be provided by the user. This value will appear selected on IPF.

### Default Value for Date type Parameter

For a date type parameter, **Default Value** will become editable dropdown. Click calendar icon to select the date. You may specify a date in it or select a value from the options:

- **CURRENT\_DATE:** The date on which report will be generated.

- **MONTH\_START\_DATE:** First day of the month in which report will be generated.
- **YEAR\_START\_DATE:** First day of the year in which report will be generated.

You can also set a default value that is relative to any of the three options listed above. For example, to set default date as 3 days after CURRENT\_DATE, CURRENT\_DATE + 3. To set default date as 5 days before month start, specify MONTH\_START\_DATE - 5.

To provide relative values as above, select the option from editable dropdown and suffix it by + or - and the number.

For example, if report print date is August 13 2009 and default date is MONTH\_START\_DATE - 5, IPF at run time will display July 27 2009 (in the selected date format).

Refer to online help for explanation of Date Time format codes.

**Time Zone** is useful when different users access application from different time zones. In such cases, it may happen that date/time data stored in database may be in one specific time zone and user may be accessing application from a different time zone. In this situation, application can convert date / time type data from one time zone to another time zone.

In **User Time Zone**, select the time zone from where user is expected to access the application and so provide parameter value in that time zone (to convert from). Select **SYS\_USER\_TZ** to use time zone applicable at run time (depending pre-set priority by the application). Select **SYS\_SERVER\_TZ** to use time zone set on **Server Properties** page (Report Server's time zone).

Check **Prompt User Time Zone** checkbox if user will select time zone at run time at the time of providing value for this parameter on IPF.

In **Database Time Zone**, select the time zone in which date / time data was entered in the database (to convert to). Select **SYS\_CONN\_TZ** to use time zone set on **Database** page. Select **SYS\_SERVER\_TZ** to use time zone set on **Server Properties** page (Report Server's time zone).

For time zone conversion to take place, value for **Database Time Zone** and **User Time Zone** needs to be provided.

**Description:** Specify information about parameter that will help user in providing the desired parameter value. Description will be displayed in a popup on IPF when user clicks the Question mark icon next to the parameter. Description is also displayed on **Save Parameter Object** dialog and **Open Parameter Object** dialog.

**Enable:** To define if this parameter should be enabled on IPF or not. By default it is checked. Report designers may choose to disable it on this page, but may enable it at report run time through scripting.

**Input Type:** This defines the way user will enter parameter value(s) on IPF:

- **TextBox:** User will type in the parameter value.

- **Combo:**
  - **Multiple values selection:** Check **MultiSelect** checkbox. Values will be listed in a box. User will click a value to select it. To select multiple values, press and hold down Ctrl key and click the values.
  - **Single value selection:** Uncheck **MultiSelect** checkbox. Values will be listed in a dropdown box. User will click a value to select it.
- **Options:**
  - **Multiple values selection:** Check **MultiSelect** checkbox. Values will be listed in a box as checkboxes. User will check values to select.
  - **Single value selection:** Uncheck **MultiSelect** checkbox. Values will be listed in a box in the form of radio buttons. User will click in the radio button to select that option.

### **Input type for Boolean type parameters**

Boolean type parameter is displayed as check. It has only two states: Checked and Unchecked. To setup a Boolean type parameter,

1. Select data type as **Boolean**.
2. Within **Value** area, in **Checked**, specify value to be passed when checked.
3. In **Unchecked**, specify value to be passed when unchecked.

At run time, parameter name will be displayed on IPF as a checkbox, which user will check or uncheck as desired.

**Mandatory:** Check to make it mandatory for user to specify value for this parameter in order to generate the report. On IPF, mandatory parameters will be listed under Mandatory tab. Other parameters (for which Mandatory is not checked) will appear on Optional tab.

**Visible:** This property defines if parameter will be visible on IPF or not. By default, this checkbox is checked. So, this parameter will be displayed. Some reports do contain parameters, but its value is passed without direct human interaction - like subreports and hyperlinked reports. If this parameter is used in such reports, then its value will be passed by the calling report. If this is the case, uncheck this checkbox. Uncheck this checkbox if you want to run report using the default parameter value only.

**Restrict to List:** This is applicable for parameters for which Input type is Combo. By default it is checked to make sure users select value(s) only from the list.

**Forced:** (available to SuperAdmin users): As a super administrator, you can configure a set of parameter values for each user. The user can thus select

value(s) from the specified set only (only these values - Forced values - will be listed in the parameter for the user).

For example, you want Tom to select values from "Central region" and "Western region", and John to select values from "Eastern region" and "Alaska region". If you check this checkbox, Tom will have to select between "Central region" and "Western region" and John will have to select between "Eastern region" and "Alaska region".

**Note:** To provide restricted access, navigate to Navigation > Administration > User / Role > Data Restriction.

**Combo Source:** This is applicable when input type for the parameter is *Combo* or *Option*. Values that should appear can be pre-defined or fetched from database using an SQL. [Click here](#) for instructions.

A parameter can also be used on report as a control (like other report fields). Check **Display Parameter Name** checkbox and specify a name. This name will appear as control name on report designer, which you can drag and drop on canvas (the report layout area). **Display Parameter Name** has no effect when the Parameter Object is used in Adhoc Report.

## Multi Select Parameters

Check the checkbox in **Multi Select** tab header if user may need to select / specify multiple values for this parameter. For example, for Country Names, user may select multiple country names.

- **Pass Values Using Tables:** Check this checkbox when you want to pass multiple parameter values through table. This is done specially when number of values that can be passed (total number of bytes of selected values) as part of stored procedure or SQL is more than what is allowed. Refer to online help to know more on this.
  - **Enclosed By:** Specify the character that would be used to enclose the set of values. This will depend on the database.
  - **Separator:** Specify the character that would be used to separate two values. This will depend on the database.
- **Maximum selectable values:** Specify the maximum number of values a parameter can take as input.

## Setting multiple default values

A multi-select parameter may have multiple default values. Default values will be displayed selected at runtime on IPF. Under Select Default Values area, select any of the following options.

- **All:** All values displayed as selected at run time.

- **Selected:** To display some of the values as selected at run time, click Selected option and select those values from the list (appearing below).

**None:** To display no value as selected at run time.

## Setting Combo Source

This is applicable when input type for the parameter is *Combo* or *Option*. Values that should appear on combo / option can be pre-defined or fetched from database using an SQL.

### To set pre-defined values

The screenshot shows a dialog box titled "Combo Source". At the top, there are two radio buttons: "Pre Defined" (which is selected) and "SQL". Below this is a section titled "Pre Defined List" containing a table with two columns: "Display Name" and "Value". Each column has an empty text input box. To the right of the table are four control buttons: a "+" button for adding items, a "-" button for deleting items, an up arrow button, and a down arrow button. At the bottom of the dialog, there is a checkbox labeled "Display Parameter Name" followed by an empty text input box.

Figure 2: Pre Defined options for a Combo Parameter

1. Under **Combo Source**, select *Pre Defined* option.
2. In **Display Name** entry box, specify the value that should be displayed on Input Parameter Form (IPF).
3. In **Value** entry box, specify the value that should be used (passed as filter).

Click button to add these values in list. Repeat the steps 2 and 3 for each option.

- To delete an option from list, select the value and click button.
- To move a value up in the list, select the value and click button.
- To move a value down in the list, select the value and click button.

## To get values from an SQL

The screenshot shows a 'Combo Source' dialog box. At the top, there are two radio buttons: 'Pre Defined' (unselected) and 'SQL' (selected). Below this is a text area labeled 'SQL' containing the text 'Select \* from Product'. To the right of the text area is a 'Verify SQL' button. Below the text area are two dropdown menus: 'Display Column' with 'PROD\_NO' selected, and 'Value Column' with 'PROD\_NO' selected. At the bottom, there is a checked checkbox labeled 'Display Parameter Name' followed by a text field containing 'ParameterObject\_disp'.

Figure 3: Defining an SQL for a Combo Parameter

1. Under **Combo Source**, select **SQL** option.
2. In **SQL**, specify SQL to be used to get data.

**Tip:** To fetch all the fields from a table, instead of writing entire SQL (for example, SELECT \* FROM PRODUCT), you may specify only table name (For example, PRODUCT).

3. Click **Verify SQL** button. If the SQL is valid, the fields it returned will be listed in **Display Column** and **Value Column**.

**Note:** If SQL is invalid, an error message will be displayed in SQL box.

4. In **Display Column**, select the field whose value should be displayed to the user. For example, CompanyName.
5. In **Value Column**, select the field whose value should be used (passed as filter). For example, CompanyCode.

**Note:** You can also specify a Hierarchical MDX as well as Drill Through MDX.

## **Setting Search**

When there are hundreds of valid values for a parameter, combo or options input type alone wont help. Search enables users to get values dynamically based on conditions that he/she sets at run time. Values selected from these options will be used as filter during report generation. [Click here](#) to know how to setup Search.

## **Setting Tree View**

Tree view displays parameter values in hierarchical form. It provides user confidence when selecting a value that may exist multiple times in the data source. For example, Dept: Finance, Office: HO and Dept: Finance, Office: WestRO. User can navigate to the desired branch to select value. [Click here](#) to know more about tree view.

## **Linking the parameters**

A report may have multiple parameters. In this case, valid value(s) for a parameter (eg. "Cities") may depend on value(s) specified in other parameter ("States"). This is achieved by linking parameters. [Click here](#) to know how you can link parameters.

## **Validations**

Applying range validations to parameter makes sure user does not key in an invalid value. Validations can be set if Input type of the parameter is Text. [Click here](#) to know how to setup validations.

## Setting Search

If a user (at run-time) enters parameter value in a text box, he/she may provide a value that does not exist in database. Offering a Combo or Options (so that user can make selection instead of entering a value) may be a good idea for only when there are limited values as options. When multiple or large values are selected, multiple selection might be difficult and time consuming.

Search enables a user to get values dynamically based on conditions that user sets at run time. Values selected from these options will be used as filter during report generation.

Search is available when parameter input value is set as Combo or Option and Combo Source is selected as SQL.

### To get filtered records on Result tab

1. Check **Search** tab header to enable it.
2. Select a column in **Column Name**. Its **Data Type** will be displayed automatically. You may select a different one if needed.
3. Specify **Prompt** that should appear on filter tab of IPF at run time.
4. Select **Operator** to set filter condition. The values displayed on operator would change according to the **Data Type**.
5. Specify the value under **Value1** (and **Value2** depending on condition) that would get records based on the specified values/condition.

**Note:** After setting filters on **Search**, when you open **Multiple Default Values** area, you will get list of values filtered based on conditions set here.

### To add a condition

Click  button available on top right of the area.

### To remove a condition

Click  button available on the right of respective row.

### At runtime

When user runs the report having this parameter, **Input Parameter Form** (IPF) will open up.

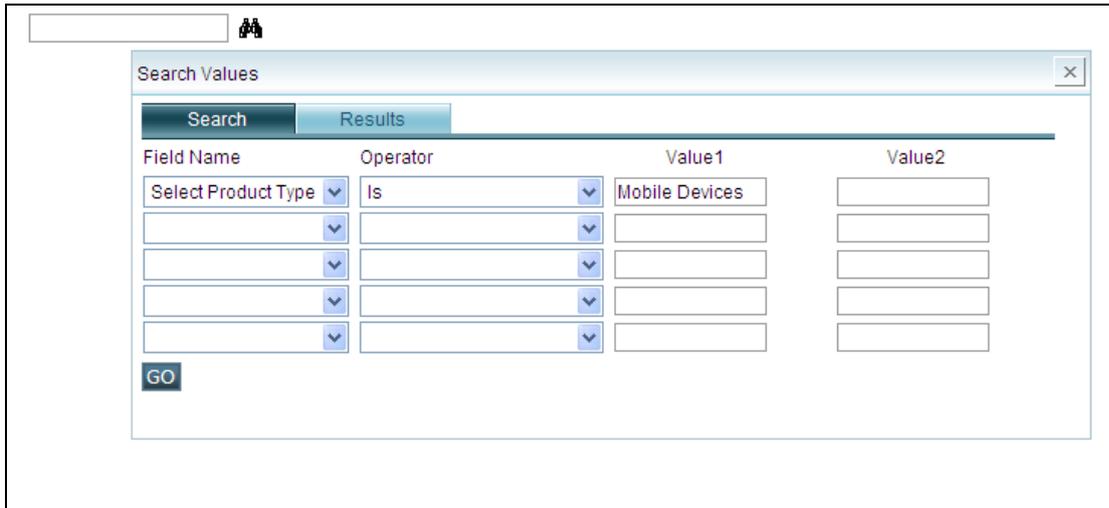


Figure 4: Partial view of Report Parameters page

It will have a binoculars (search) icon for this parameter. User will click the icon to get **Search Values** dialog box.

Specify values on **Search Values** dialog box's **Search** tab and click **Go** button. **Result** tab will display values belonging to the records that satisfied the filter criteria provided on **Search** tab.

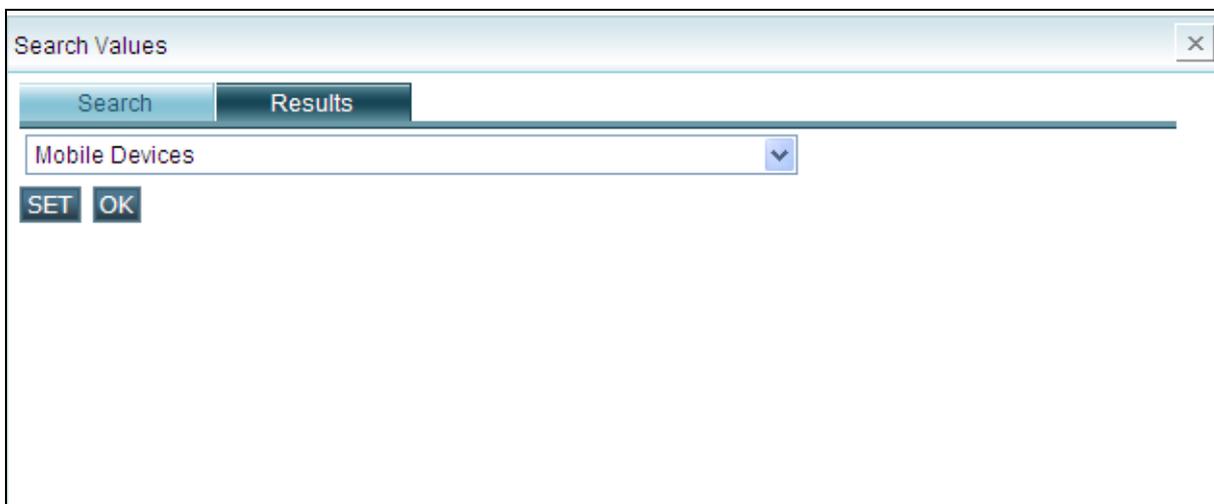


Figure 5: Values received based on Search Criteria

User will select desired value(s) from **Results** tab and click **Set**, followed by clicking **OK**. This dialog box will be closed and selected value(s) will be listed on IPF.



Figure 6: Values are listed

Click **Run Report** to run the report filtered by these values.



**Note:** Tree view is not available for Search Options.

## Tree View

Tree view represents parameter values in hierarchical form. This provides the user more information about the parameter value.

Example: Cities.

### Text box view

- Indore
- Bhopal
- Bangalore
- Los Angeles
- San Francisco
- Las Vegas

### Tree View

- India
  - Madhya Pradesh
    - Indore
    - Bhopal
  - Karnataka
    - Bangalore
- United States (parent node)
  - Nevada (child node)
    - Las Vegas (leaf)
  - California
    - San Francisco
    - Los Angeles

In this view, while users are able to view the parameter values (as last item in the hierarchy - leaf), they are also able to view other information about the parameter value. In this example, users are able to view the state and country in which a city is located.

In addition to above, on Input Parameter Form, users can select a branch and move all its value to at a time.

Check **Tree View** tab header checkbox to enable the area and work on it.

Following type of tree views are available to choose from:

- **Flat:** Select this when you are using a flat database structure. When Flat is selected, the leaf is set as Display Column under Combo Source. Nodes are set under Tree View Details. For example, for a three level tree, you will set two levels in Tree View Details and third in Combo Source. Make sure the SQL used to get parameter retrieves all the fields required to create the tree.
- **Hierarchical:** Select this when the database has hierarchical relationship. In Oracle, for example, when database has hierarchical relationship, the SQL uses Start with and connect by clause.

In case of Hierarchical, the query should return:

- **NODEID:** Unique identification value of the node.
- **PARENTNODEID:** Unique identification value of current node.
- **NODLEVEL:** A number indicating node level of current node.
- **NODEVALUE:** Actual value of node.

Example query for Oracle

```
select      child      "NODEID",parent      "PARENTNODEID",level
"NODELEVEL",child "NODEVALUE"
from test_connect_by
start with parent is null
connect by prior child = parent
```

When **Flat** is selected,

1. In the first row of LEVELS, select the field that should appear topmost in the tree view.
2. To have second branch, set the fields in second row.

Similarly, add more branches if required. Click  button to add a row. Click  to delete respective row.

At report run time, dual list will be presented to the user to select from. User will be able to select

- One or multiple child nodes under a parent (for 'multi-select' parameters)
- Parent node along with child nodes
- Parent node without having need to select child node

**Note:** Tree view is not available for Search.

## Linked Parameters

A report may need values from multiple parameters. In this case, valid value(s) for a parameter may depend on value(s) specified in other parameter.

For example, valid values for "Cities" will depend on value(s) selected in "States". You may link both the (parent and dependent) parameters. When you link the parameters, values in "dependent" parameter ("Cities", in example here) will be populated depending on "parent" parameter ("States", in example here).

### General steps to get filtered list by Linking parameters

1. Select **Linked Parameter** tab header.
2. Check the parameters to be linked (parent parameter) with the parameter being created (dependent parameter).
3. For the SQL being created for SQL Combo Source, specify name of the parameters being linked enclosed by <% and %>.

For example, to get list of belonging to state selected in another parameter, specify this SQL in SQL box of this parameter:

```
Select CityNM from FinData where FinState=<%State%>
```

At run time, Input Parameter Form will appear having these two parameters. User will specify value for State, which will be used in the SQL to fetch values (for example CityNM) for this parameter. This combo will have only the cities belonging to the selected State.

## Validations

Applying range validations to parameter makes sure user does not key in an invalid value. Validations can be set if Input type of the parameter is Text box. To enable the validation tab, check **Validation** checkbox on the tab header.

You can provide:

- Valid values (Characters, numbers of dates)
- Invalid values (Characters, numbers of dates)
- Script to be executed at run time to validate the entered value.

### Specifying validation for Number type parameter

To specify **Range of numbers**, mention starting number in **From** box and ending number **To** box (of the same row).

If valid value is a number onwards, mention in the starting number in **From** box, leave **To** box blank.

If a valid value is up to a number, leave **From** box blank and mention the ending number in **To** box.

You can specify positive numbers and negative numbers too. A number may or may not have decimal point.

Click  button to insert a row below. Click  button to delete the row.

### Specifying validation for Character type parameter

To mention **Allowed Characters**, you may type in the characters or select the characters from Character set dialog.

#### Selecting characters from Character Set dialog

Click ... button to open **Character Set** dialog. Click a character to select it. Selected characters appear depressed. To unselect a character, click it once more. After making the selection, click **OK** to close the dialog and return to **Parameter Detail** dialog.

#### Range(s)

Each character has a unicode 'number'. Unicode of the character will be considered for range validation.

To specify **Range of characters**, mention unicode of starting character in **From** box and that of ending character in **To** box (of the same row).

If valid value is a character onwards, mention unicode of starting character in **From** box, leave **To** box blank.

If valid value is up to a character, leave **From** box blank and mention unicode of ending character in **To** box.

Click  button to insert a row below. Click  button to delete the row.

## How to specify Date values

You can specify a date or from dropdown, select any one among

- CURRENT\_DATE (The date on which the report would be generated).
- MONTH\_START\_DATE (First day of the month in which the report would be generated).
- YEAR\_START\_DATE (First day of the year in which the report would be generated).

## Range(s)

To specify **range of dates**, mention date in **From** box and **To** box (of the same row).

If valid value is a date onwards, mention date in **From** box, leave **To** box blank.

If valid value is up to a date, leave **From** box blank and mention date in **To** box.

Click  button to insert a row below. Click  button to delete the row.

## About Scripts for parameter value validation

You can add a validation script for a parameter. Click **Add Script** button to open **Script Editor** dialog and write the script.

At parameter level, OnChange() event is supported. It means, validation script will be executed when:

1. User types in a value for the parameter (for input type TEXT), or
2. Selects/Unselects value from the parameter combo/list/tree.
3. Checks/ Un-checks a check box.

Validation script written at parameter level can access other report parameters. It can also access parameter objects (even if not imported) and global business parameters. This will be Read-only access.

If the parameter value is valid, script will return True. If it is invalid, script will return False. You can set an error message that should be displayed if parameter validation fails. Report will not be generated if parameter validation fails.

Using script, you can modify attributes of parameters. (For example, if paramA is invalid, disable paramB.) IPF will reload parameters that are affected by the script.

In case of scheduled report execution, IPF is not displayed. Hence, script will be executed at the time of saving of schedule tasks. Script will not be executed at report run time.

### Modifying a Parameter Object

On Parameter page,

1. Open the parameter object.
2. Make changes where required.
3. Click **Save** button to save the changes.

### Deleting a Parameter Object

On Parameter page,

1. Open the parameter object.
2. Click **Delete** button.
3. When Alert is displayed, click **OK**.

The PO will be deleted.

**Important:** When you delete a PO, all the reports where it is used, will fail to execute.

### Parameter Objects from Data Model

You can access a PO from Data Model. Data Model Explorer not only provides hierarchical view of folders and Parameter Objects within each folder, it also allows you to carry out many operations on selected PO.

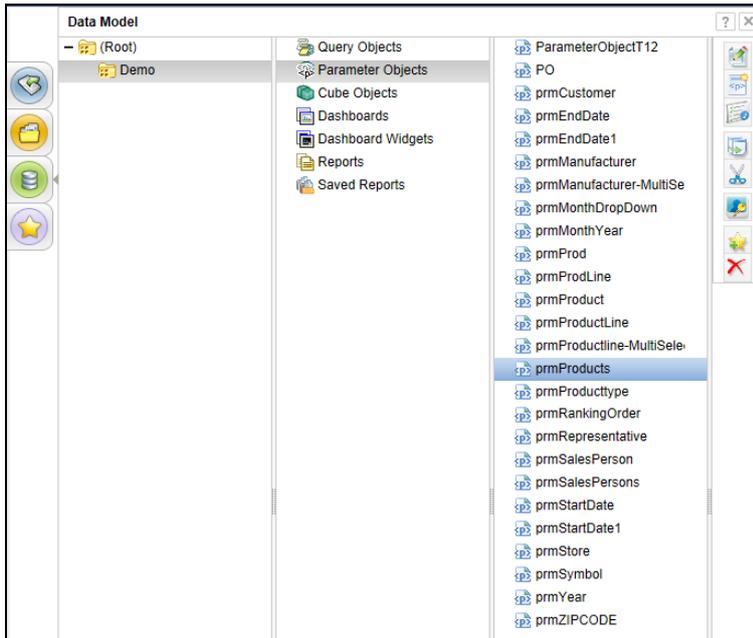


Figure 7: Parameter Objects on Data Model Explorer

You can do following when you select multiple PO at a time:

- Set Access Rights
- Copy, paste and cut paste the PO
- Delete the PO

If you select only one PO, you can also do following in addition to the above:

- Edit the selected PO
- Add the selected PO to favorites

Refer to online help to know more on for each of these actions.

You can create a parameter Object by navigating to a parameter object from here also.

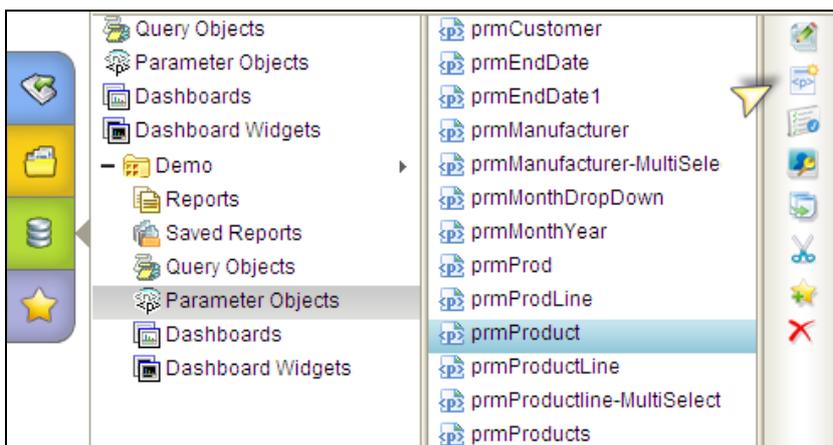


Figure 8: Create Parameter Object from Repository

You can create a parameter Object by navigating to a parameter object from Favorites tab also.

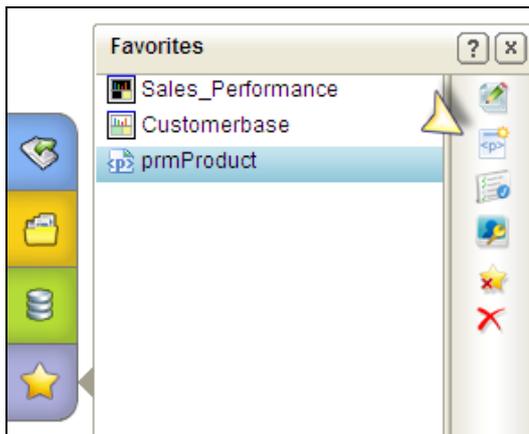


Figure 9: Create Parameter Object from Favorites

## Parameter Value Groups

Some of the reports may need users to provide multiple values at run time, for example, Country. Selecting a few country names from a long list may be very difficult.

On this page, admin can create parameter value groups. Examples: Americas (having countries in North American sub-continent), Europe (having countries in Europe), Asia, Africa and specify values in that group. At run time, when a user selects a group, values belonging to that group will appear as selected. User does not have to manually select each of the countries every time user runs the report. This saves time as well as reduces chances of errors.

The parameter Value Groups page can be accessed from:

- Data Model > Category Name > Parameter Objects > Parameter > Parameter Value Groups
- Navigation Menu > Repository > Report Objects > Parameter Value Groups

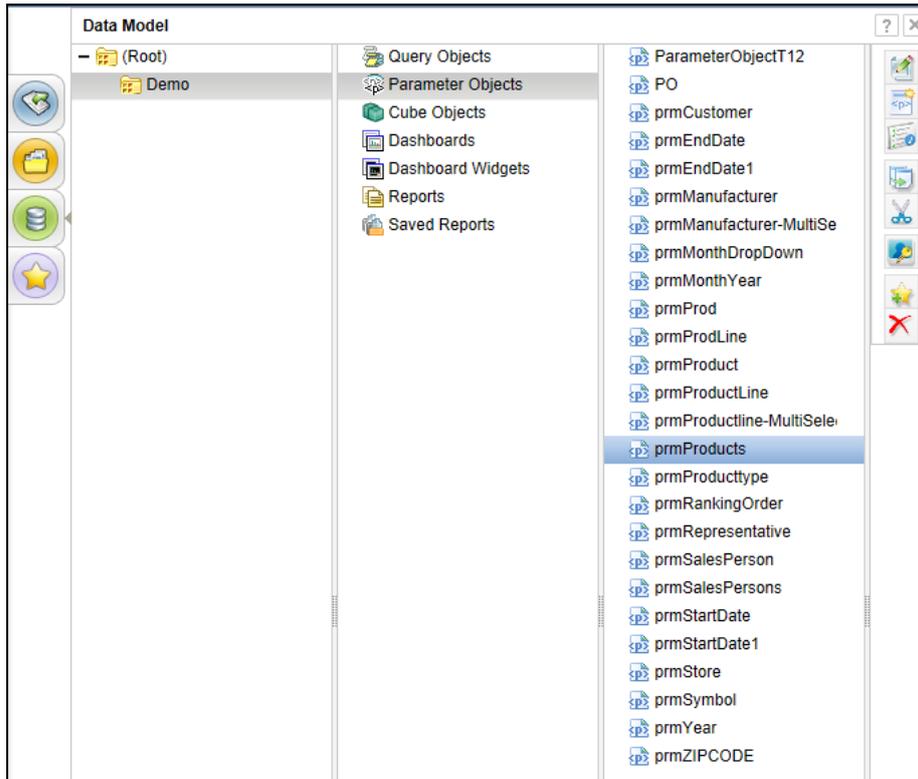


Figure 10: Parameter Value Groups 1

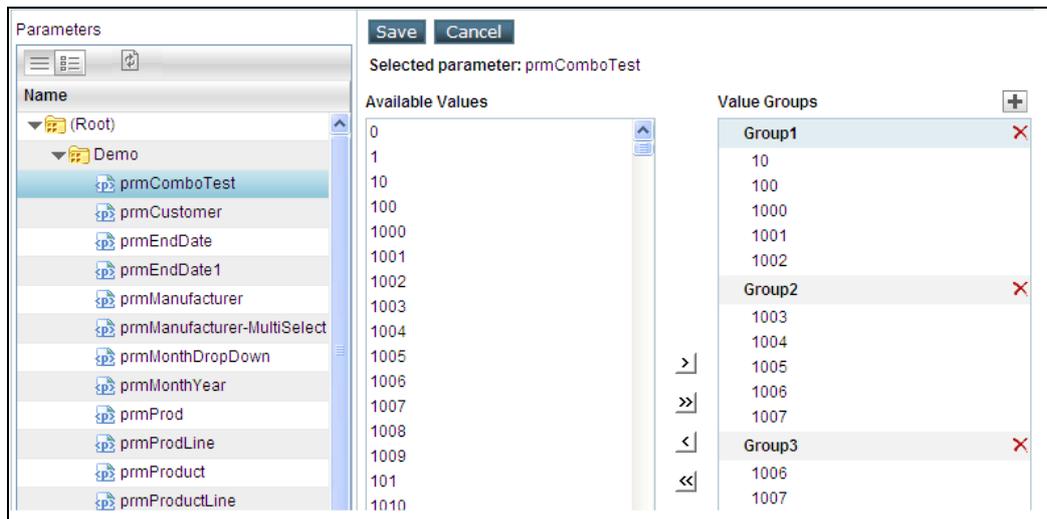


Figure 11: Parameter Value Groups 2

The page has following information:

- **Parameters:** Lists all the parameter objects.
- **Available Values:** Lists available values for the selected parameter.
- **Value Groups:** Lists groups created and the values selected within a group. An icon appears on the left of a Private group.
- **Show All Owners:** If selected, displays groups created by all the users. Such groups will have  icon in the group title.

- **Option buttons Private and Public:** Select *Private* if user wishes to list the groups that you have set for you only. Select *Public* if you wish to list the groups that you have set for everyone.

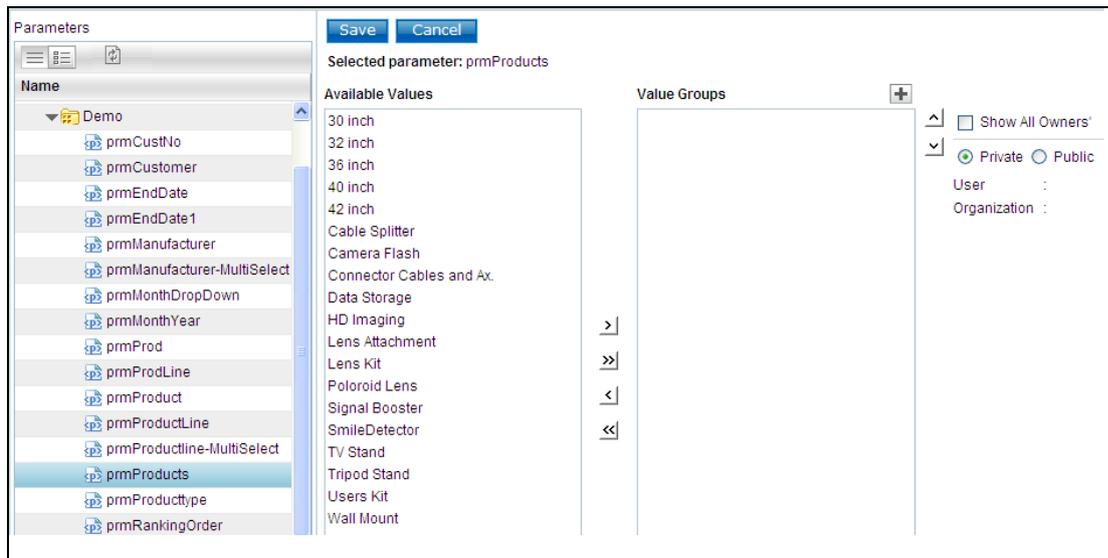


Figure 12: Creating Parameter Value Groups

## To create group

1. Click **Plus** button available above **Available Values** box. Group is created.
2. Double-click the name to open it for changing the name of group. Specify the name and click outside the box.
3. Add the values in the group by selecting a value in **Available Values** list and clicking **>** button. Selected value is added under the selected group in **Value Groups** box.
4. Repeat step 3 for each value to be selected.

If a value that you want to add in the group is not listed in **Available Values** list, specify the value in **Additional Value** entry box and click **>** button.

Use **>>** button to add all the values, **<** button to remove selected value, and **<<** to remove all the values from **Value Groups** box.

Select a group and click **▲** or **▼** arrows to move selected group up or down. Select a value and click **▲** or **▼** arrows to move the selected value up or down (within the group).

Click **Save** button to save the work.



**Note:** Please note that if name of a group that is used by a user, is changed, the values under that group will be removed from Selected Values group of that user's preferences.

### For Tree view parameter

To select a value, click the leaf node and click  button.

To select all values in a branch, (only for a multi-select parameter) click respective branch and click  button. All the values under that branch will be selected.

**To make changes in name of a group:** Double-click the group name. It opens for modification. Specify new name and click outside the box.

**To delete a group:** Click  in the title of respective group.

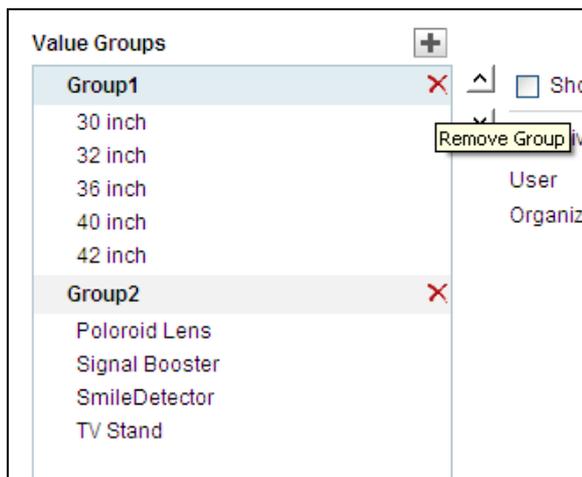


Figure 13: Deleting a group

Click **Save** button to save the changes.

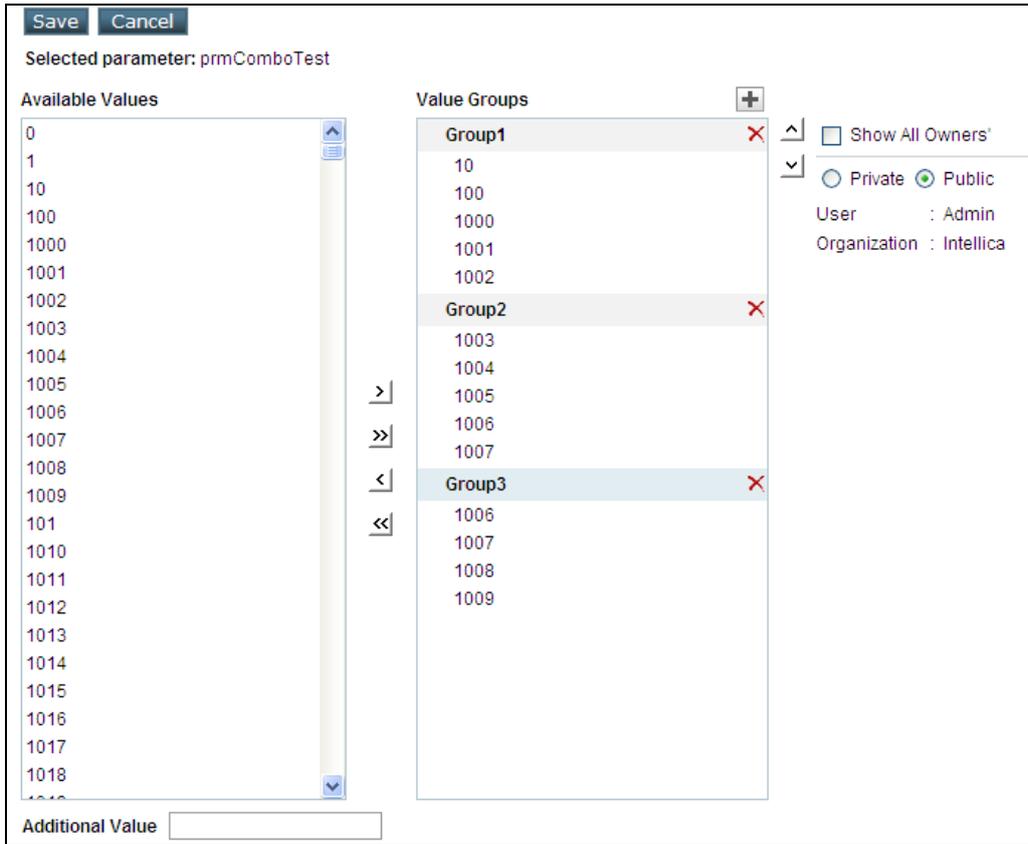


Figure 14: Parameter Value Groups page is displayed