



---

# Desktop Studio: Report Parameters

---

Intellicus Enterprise Reporting and BI Platform



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

---

Copyright © **2010** 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: - October 2012**

## **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>Working with Parameters .....</b>	<b>4</b>
Parameters List Dialog .....	4
Importing parameters from repository .....	5
Creating New Parameter .....	6
Linked parameters .....	20
Parameter Validations .....	20
Exporting a parameter .....	23
Parameters Form Layout .....	24
Using Report Parameters in SQL Editor .....	26
Using Report Parameters in Layout Editor .....	26
Using System Parameters .....	26

---

## Working with Parameters

---

Report Parameters are the inputs provided by end user before the execution of reports.

Intellicus Studio allows you to design parameters and use them in reports. Studio also allows you to import or refer to parameters from the repository. Generally the parameters are used in the report SQL to apply appropriate filters.

### Parameters List Dialog

The **Parameters** list dialog is used to add, delete, or import parameters. You can import or import by reference from repository. You can also create new parameter object in repository using the definition created in local report for using it in other reports.

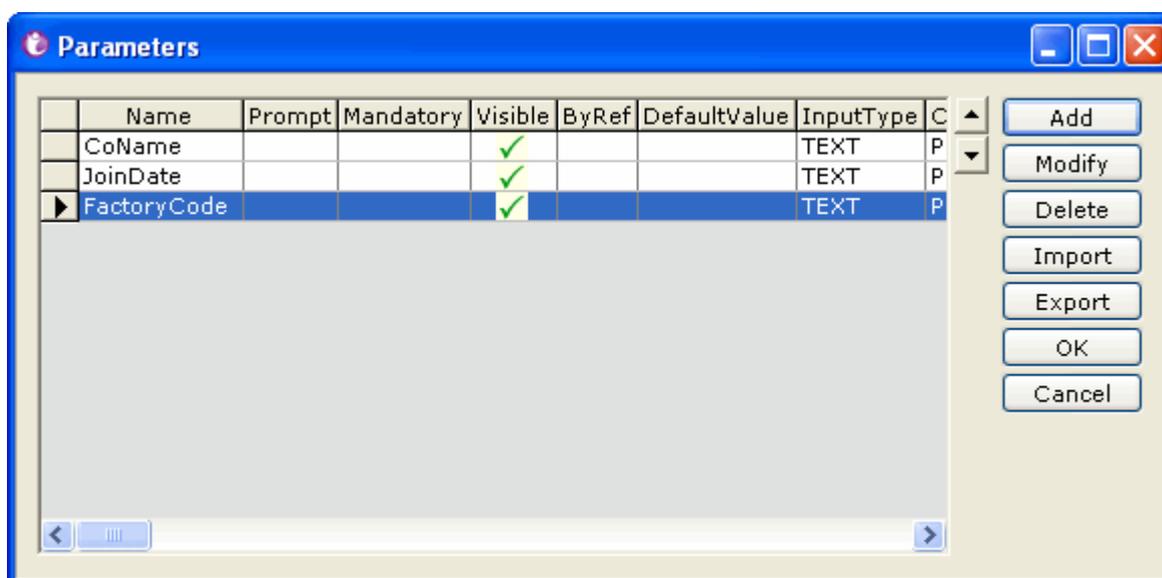


Figure 1: Report Parameters Dialog Box

#### Actions:

Action	Comments
Add	Add a new parameter with local definition. This parameter will be local to this report only.
Modify	Edit properties of local parameter. This option will not be available for parameters imported from repository by reference
Delete	Delete selected parameter from this report. If it is locally created parameter, then its definition will be deleted. If it is an imported parameter, local copy or reference is deleted and the original copy remains in the repository.
Import	Import one or more parameters from repository. ByRef: Yes/No By reference – Yes = The latest definition of the parameter object will be used at the report run time

	By reference – No = The definition of the parameter object at this time will be copied from repository and a local copy is created. Any changes in repository parameter object will not affect this report. See Dialog image below this table
Export	This option is available if the user has Administrator privilege or access rights to create parameters objects in the repository. A new parameter object is created in the server repository with the definition of the local parameter.
OK	Saves all changes made to parameter list in the report.
Cancel	Close dialog without making changes to the report. Changes made to repository by Export cannot be reverted by Cancel.
Up and Down	The sequence of parameters appearing on the parameter form is decided by the sequence of parameters seen in this dialog. You can move the parameters up and down to achieve the desired sequence of parameter prompting on the parameter form.

### Parameter Import Dialog:

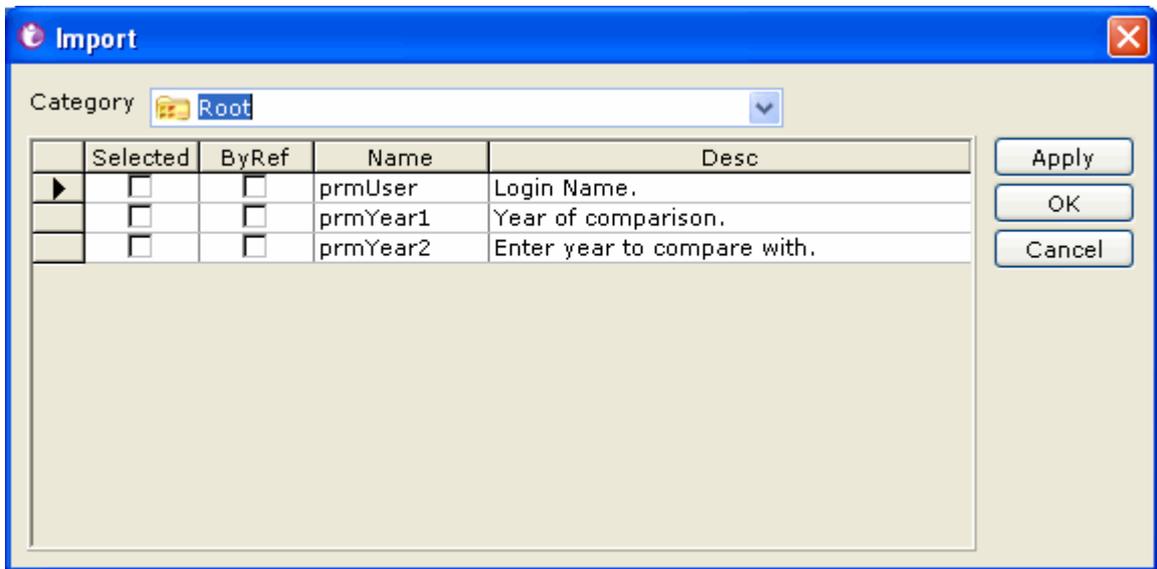


Figure 2: Importing a parameter

### Importing parameters from repository

1. Click **Import** button on the **Parameters** dialog box. The **Import** dialog box opens.
2. In **Category** object selector helps you navigate to folder to list the parameters below.
3. To select a parameter for import, check the **Selected** checkbox in corresponding row.

4. To select a parameter to be imported by reference, select the check box By Ref in the same row.
5. Click the **Apply** button.

Selected parameters are imported to the open report. You may click **OK** to import the parameters and also close the dialog.

## Creating New Parameter

To create parameter, click menu Tools → Parameters to open the **Parameters** dialog box. On **Parameters** dialog box, click **Add** button.

The options in this dialog box are given below.

### **Name:**

Specify a name to uniquely identify this parameter. You can access this parameter in the report using this name enclosed by <% and %>

### **Prompt:**

Specify the prompt caption that should appear on input parameter form while prompting for this parameter.

### **Data Type:**

Select the type of value user will provide at run time. Data type of a parameter decides the tools provided and validations applied while user entering the values for this parameter.

<b>Data type</b>	<b>Input tools</b>	<b>Validations</b>
CHAR	Text box, List Box Multi select List box	Alphabets, Numbers or special characters are allowed
NUMBER	Text Box	Numbers are allowed
DATE	Date, Date part, Variable date	A valid date according to selected input format. Predefined data variables such as MONTH_START_DATE YEAR_START_DATE CURRENT_DATE
BOOLEAN	Check Box	Check or Uncheck

### **Size:**

Specify number of digits or characters that this parameter value can accept.

### **Format:**

Select the input format in which user should provide value for this parameter. Use the  button to open **Data Format** dialog box. Based on the format you have selected, a format string will appear in the entry box.

Format string characters for date and time:

Characters	Use	Example (Friday, December 26, 2008, 17:46:13 hours)
<b>dd</b>	Date	26
<b>ddd</b>	Day in 3 characters	Fri
<b>dddd</b>	Complete day name	Friday
<b>hh</b>	Hour of time in 2 digits	17 (or 05)
<b>Mm</b>	Minute of time in 2 digits	46
<b>MM</b>	Month in number in 2 digits	12
<b>MMM</b>	Month name in 3 characters	Dec
<b>MMMM</b>	Complete month name	December
<b>ss</b>	Seconds in 2 digits	13
<b>yy</b>	Year in 2 digits	08
<b>yyyy</b>	Year in 4 digits	2008
<b>a</b>	time in 12 hours format	- -

### Negative numbers

You can provide format strings (patterns) for positive number and negative number. To separate both the patterns, use a semi colon (;) character. For example,

`##,###.##;(##,###.##)`



**Note:** Negative pattern is optional. If you omit negative pattern, localized minus sign (-) will be considered to represent a negative number.

## Default Value:

Default value comes pre-populated on the parameter control on the input parameter form when it is loaded. For multi select type controls you can select multiple default values.

The screenshot shows the 'Parameter Details' dialog box. The 'Name' field is 'ProdNo', 'Prompt' is 'Product Name', 'Data Type' is 'NUMBER', 'Size' is '5', and 'Default Value' is '40100,40101,40102,4'. The 'Input Type' is 'Combo'. The 'Multi-Select' checkbox is checked. The 'Select Default Values' section has 'Selected' selected, and a list of products is shown with checkboxes: Audiovox XV-6600, DG VX6100, PCS Vision PPC-6601, Nextel i325, Mini Case Protectors, Pink Color Cancer Cellphone, Extra Large Case Protectors, and Over Ear For 2.5mm Phones. The 'Combo Source' section has 'SQL' selected, and the SQL editor contains the query 'select \* from product'. The 'Display Column' is 'PRODUCT' and the 'Value Column' is 'PROD\_NO'. The 'OK' and 'Cancel' buttons are at the bottom right.

Figure 3: Parameter Details Dialog Box

## Default Value for Date type Parameter

When you select parameter for data type as Date, the Default Value entry box changes to a dropdown box.

Name	ProdNo	
Prompt	Product Name	
Data Type	DATE	Size 5
Format	M/d/yy	
Default Value	<input type="text"/> <input type="button" value="..."/>	
Database Time Zone	CURRENT_DATE	<input type="button" value="v"/>
User Time Zone	MONTH_START_DATE	<input type="button" value="v"/>
	YEAR_START_DATE	
	<input type="checkbox"/> Prompt User Time Zone	

Figure 4: Options for default value of date parameter

You can specify a value in this dropdown box as well as select a value from the options:

- CURRENT\_DATE
- MONTH\_START\_DATE
- YEAR\_START\_DATE

You can take the default date ahead / behind the date falling on any of the options set. For example, you can set default date as 3 days after CURRENT\_DATE, or 5 days before month start. You can do that by specifying CURRENT\_DATE + 3 and MONTH\_START\_DATE - 5 respectively.

At run time the date that will appear will have default date accordingly.

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: Yes/No**

Enable decides the parameter is active on the input parameter form to take user input. By default it is checked.

This property can also be changed using scripting added to parameter form.

### **Input Type:**

Input type decides the input control provided to end user to enter parameter value:

Input Type	Control
Textbox	Textbox control with type in cursor
Combo	Single Select list box Multi select list box
Option	Radio buttons

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

Boolean type parameters are represented as check boxes and have only two states: Checked and Clear.

To setup a Boolean type parameter,

1. Select data type as *Boolean*.
2. Within the **Value** area's **Checked** field, specify value to be passed when it is checked.
3. In **Unchecked** specify value to be passed when it is not checked.

### **Mandatory:**

This makes a value to be given mandatorily to this parameter before submitting input parameter form. An error message pops up if end user tries to submit the form without entering any value for this parameter.

### **Visible:**

This property decides a parameter and its values are visible to end user or not. By default, this checkbox is checked. So, this parameter will be displayed. Parameters to drill down reports are defined as visible false. Data security parameters must be defined with visible false.

### **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:**

This property is visible to Super Admin users only.

As a super administrator, you can assign a set of parameter values for each user or an organization.

The user can thus select value(s) from the assigned set only.

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", then you set forced property true for this parameter and assign respective values in user management screen->data security dialog.

**Note:** Forced is enabled only when you are working with a parameter which is imported by reference. To make any changes in forced values, you need to open this parameter through Portal's Parameter page.

### Setting Combo Source

This is applicable when input type for the parameter is Combo or Option. Values that should appear, can be Pre defined or received from database using an SQL.

#### To set pre-defined values

Display	Value
Tokyo	4
New York	1
Jakarta	2
Singapore	3

Figure 5: Pre Defined options for a Combo Parameter

1. Under **Combo Source**, select *Pre Defined* option.
2. In **Display** entry box, specify the value that should be displayed to the user at run time.
3. In **Value** entry box, specify the value that should be used (passed as filter).
4. Click button to add it in list. Repeat the steps 2 and 3 for each option.

Select **Display Parameter Name** checkbox, if user may use the parameter as a control on a report.



**Note:** Display Parameter Name has no effect when the Parameter Object is used in Adhoc Report.

- 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 the 'Combo Source' dialog box. At the top, there are two radio buttons: 'Pre Defined' (unselected) and 'SQL' (selected). Below this is a large text area labeled 'SQL' containing the query 'select \* from product'. To the right of the text area are two buttons: 'SQL Editor' and 'Verify SQL'. Below the text area are two dropdown menus: 'Display Column' (set to 'PRODUCT') and 'Value Column' (set to 'PROD\_NO'). At the bottom left, there is a checkbox labeled 'Display Parameter Name' which is currently unchecked.

Figure 6: 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. To create SQL on SQL Editor, click **SQL Editor** button.
3. After having specified the SQL, click **Verify SQL** button. If the SQL is valid, the fields it returned will be listed in **Display Column** and **Value Column**.
4. In **Display Column**, select the field whose value should be displayed to the user.
5. In **Value Column**, select the field whose value should be used (passed as filter).

Select **Display Parameter Name**, if user may use the parameter as a control on a report.



**Note:** **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.

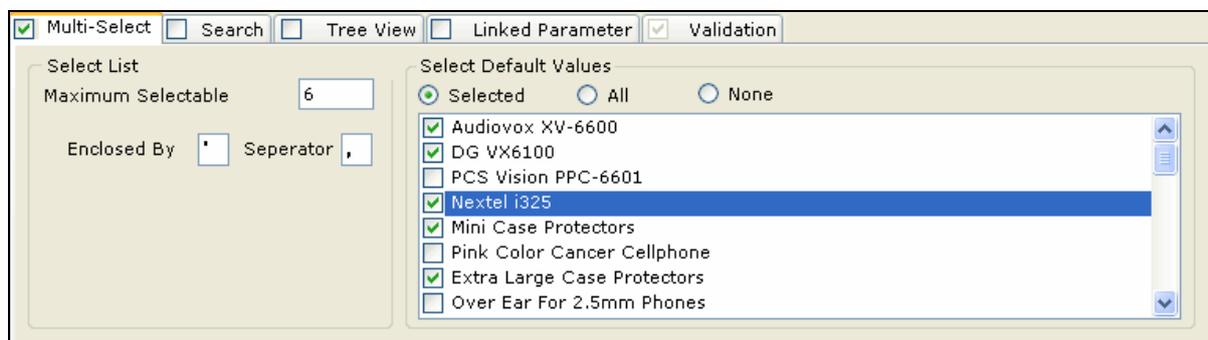


Figure 7: Multi Select Parameters

- **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.
  - **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.



Figure 8: Selecting Multi-select Default values

- **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.

## Search

If a user (at run-time) specifies parameter in a text box, there are chances that he/she may provide a value that does not exist in the database. So, you can make the parameter offering all possible values as a combo-box or multi-select options. When you have many (hundreds) of possible values, selecting becomes difficult.

You need to allow the user to filter the value that he / she wants to be offered to select from. For example, from all the product numbers, user should be able to select "product numbers of the products belonging to this category and that product line".

This is made possible through **Search Options**. Setup a parameter with **Input Type** as *Combo* and **Combo Source** as *SQL*. Specify SQL that will be used to get the record-set. Click **Search Options** button.

Here, the fields that the user will use to apply filter criteria (to get list of options in the combo) are specified.

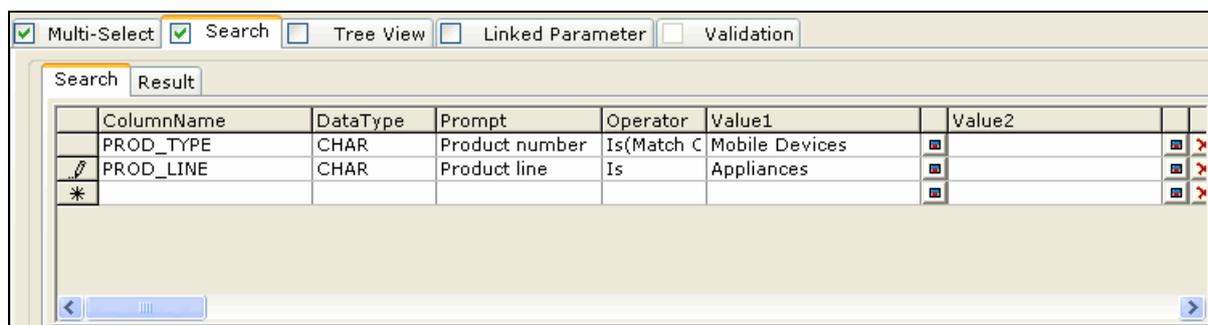


Figure 9: Setting up Search Options for user parameter

## To get filtered records on Result tab

Make sure **Searchable** checkbox is selected. Now,

1. After selecting the right field for **Display Column** and **Value Column** under **Combo Source**, click **Search Options** button to expand the area.
2. Select a column in **Column Name**.
3. Select its **Data Type**.
4. Select **Prompt** Text that should appear at run time.
5. Select **Operator** to set filter condition and provide **Value1** (and **Value2** based on condition).

An empty row will be auto-appended once you complete with present level entry. Click  to delete respective row.



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

At run time, you will be presented with the **Report Parameters** dialog box.

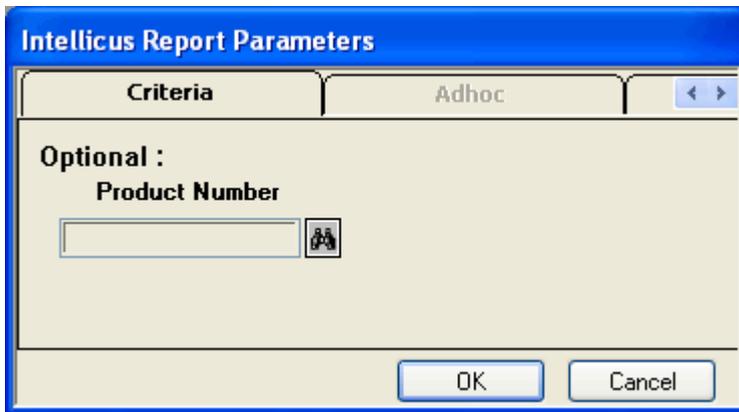


Figure 10: Report Parameters dialog box

User will click **Search** button on this dialog box to get **Search** dialog box.

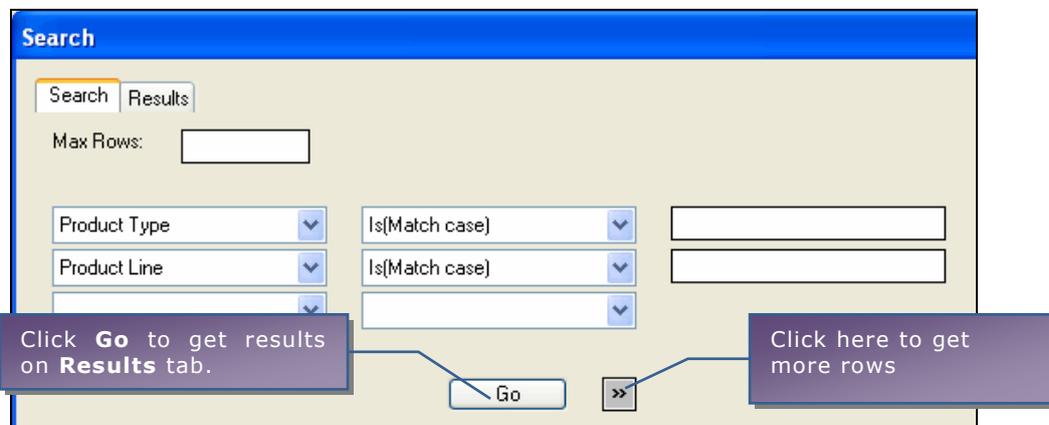


Figure 11: Specify values to filter values

On **Search** dialog box's **Search** tab, you will provide filter options. Number of results displayed on the **Result** tab will depend on the value specified in **Max Rows**.

On **Results** tab, you will be presented with only those values that satisfied the filter criteria given on the **Search** tab. You should select one value and click **Ok** to proceed.

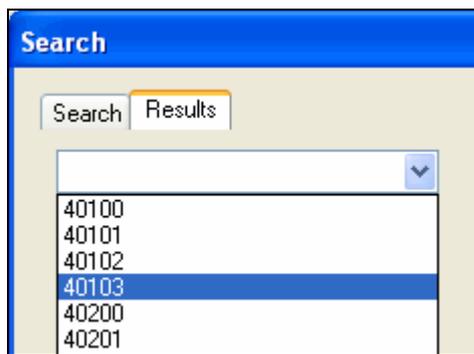


Figure 12: User should select a value.

## 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.

Input Parameter Form, users can select a branch to select all the values within the branch.

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,

Check **Tree View** tab header. For **Source Type**, select *Flat* or *Hierarchical* depending on database.

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

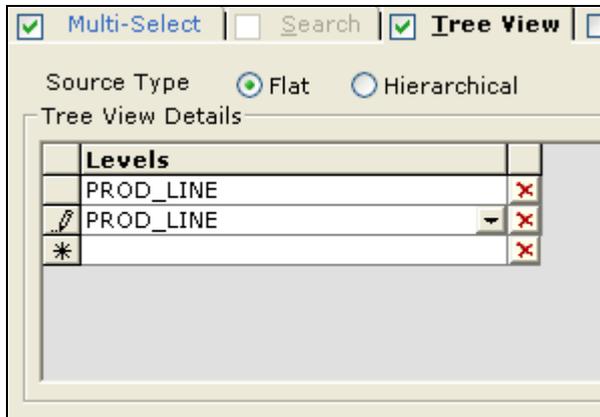


Figure 13: Setting tree view option

An empty row will be auto-appended once you complete with present level entry. Click  to delete respective row.

At report's run time, following dialog box will appear for the user to select parameters.

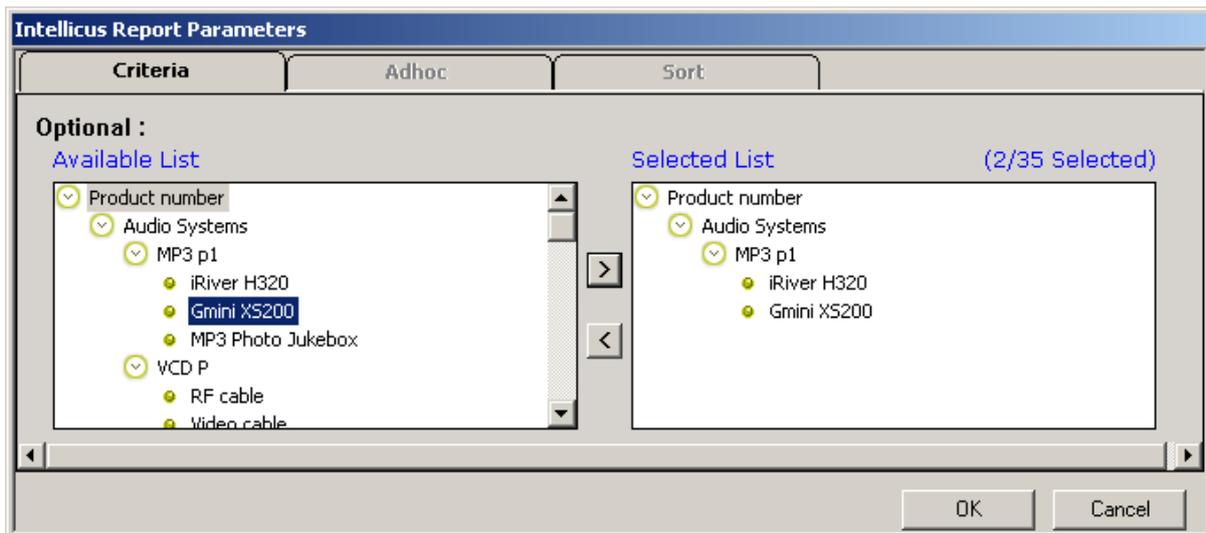


Figure 14: Tree view appears like this at run time



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

---

## Linked parameters

A report may need multiple parameter values. In this case, value displayed for a parameter may depend on value specified in other parameter. For example, values displayed in "Cities" will depend on value selected in "States".

You can link a parameter with a SQL combo type parameter.

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

1. Select **Linked Parameter** tab header.
2. Select the parameters to be linked with the parameter being created.
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 cities 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, Parameter Input form will appear having these two parameters. At that time user will specify value for State. This value will be used to fetch values for this (for example CityNM) parameter. This combo will have only the cities belonging to the value provided for State.

## Parameter 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 or dates)
- Invalid values (Characters, numbers or 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.

---

A number can be positive, negative, with or without decimal point. For example, 24, -17 and 56.77.

An empty row will be auto-appended once you complete with present level entry. Click  to delete respective 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.

An empty row will be auto-appended once you complete with present level entry. Click  to delete respective 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.

An empty row will be auto-appended once you complete with present level entry. Click  to delete respective 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 (parameter name will appear in the section), 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.

---

## Exporting a parameter

On **Parameters** dialog box, select the parameter that you want to export. Click the **Export** button. **Export** dialog box opens up. Specify the name you want to give to the parameter. Click the **Export** button. The parameter gets exported.

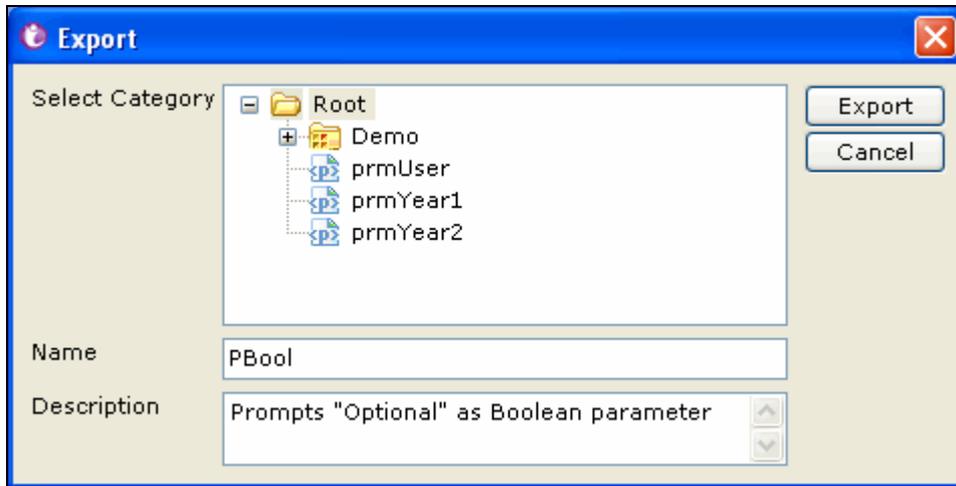


Figure 15: Exporting a parameter



**Note:** Exported parameter is stored in the Repository and is made available for all the reports.

---

## Parameters Form Layout

When a report has run time parameter(s), **Input Parameters Form (IPF)** appears on screen at report run time. Layout of the IPF can be set on **Parameters Form Layout** dialog.

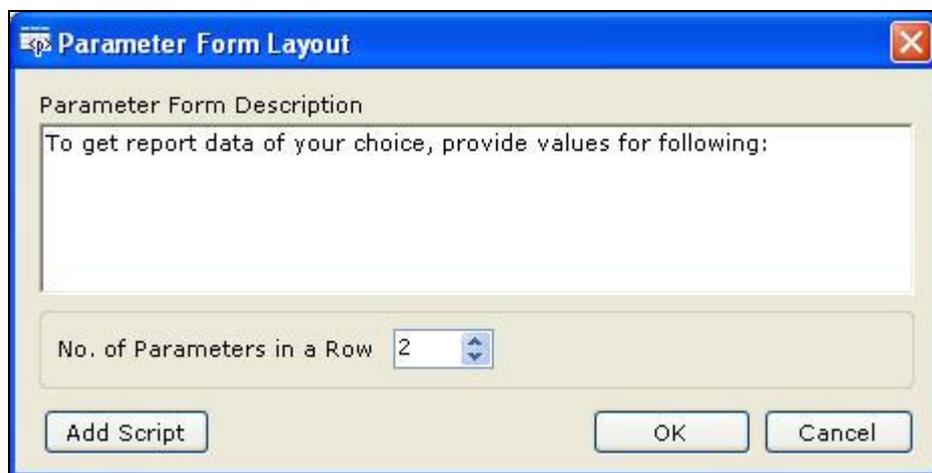


Figure 16: Parameters Form Layout dialog

To get this dialog,

- On menu bar, click **Tools > Parameters Form Layout** option.
- On toolbar, click  button.

Use this dialog to specify:

- **Description:** Description that should appear on Input Parameter Form when it is displayed at runtime.
- **No. of Parameters in a row:** Number of parameters that should appear in a row on Input Parameter Form.

Enter 0 (Zero) for auto adjusting parameter form.

Click **Add Script** button to open **Script Editor** dialog and write scripts.

At IPF level (Form level), OnSubmit() event is supported. It means, script is executed when user clicks OK / Run button on IPF.

If such a report is scheduled, IPF is presented at the time of setting the schedule and script will be executed at the time of scheduling.

Script can access any parameter of the report. This includes parameter objects (even if not imported) and global business parameters. This will be Read-only access (parameter objects and global business parameters).

If all the parameter values are valid, it will return True. If one parameter value is invalid, it will return false and report will not be executed. You can set an error message that should be displayed to the user in parameter value is invalid.

**Note:** In case of JavaScript error, Report Server will respond with ERROR.

Click **OK** to save the changes and close the dialog. Click **Cancel** to abandon the changes and close the dialog.

When you preview the report in Studio having 3 or more parameters, Input parameter form having up to 3 parameters in a row will appear.

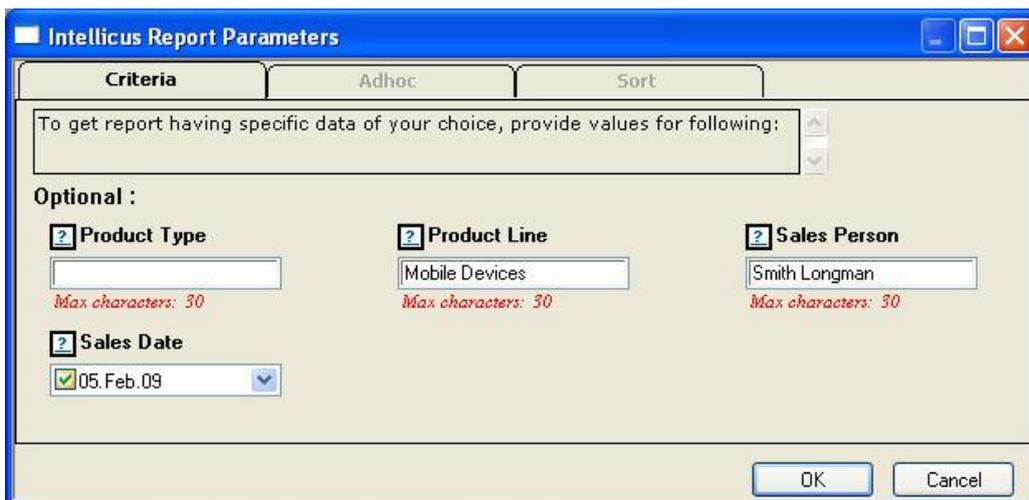


Figure 17: IPF having description and 3 parameters in a row.

---

## Using Report Parameters in SQL Editor

The parameter that you defined through the parameter window can be used under multiple options as required. In SQL editor you can use this parameter along with the column list as long as it conforms to the SQL syntax.



**Important:** Make sure that you use the parameter in the SQL enclosed between '<%>' and '%>'.

When you execute this SQL, the supplied parameter values will replace the parameter defined between '<%>' and '>%>' symbols. It is important to use single quotes while using these parameters. For example:

```
Select * from table1 where name = '<%parameter%>'
```

## Using Report Parameters in Layout Editor

You can also use the defined parameters through the field window of the Layout Editor. To use the parameters, refresh this window (View → Refresh Field List) and drag the parameter fields to the Report Layout window in the Layout Editor.

## Using System Parameters

You can use system parameters to display date and time, or other values on the report. Previously defined parameters will be visible in the field list window of the Layout Editor. You can simply drag them on the layout window.

The system parameters are global parameters that can be used in any report, as per the requirements.

Apart from the parameters defined by you, there a set of other system parameters that Studio supports:

- SYS\_DATE: Provides system date.
- SYS\_TIME: Provides system time.
- SYS\_LOCALE: Provides report output language.
- SYS\_REPORT\_FORMAT: Provides Report output format.
- SYS\_REPORT\_ID: Report ID of currently executing report.
- SYS\_CATEGORY\_ID: Provides the category under which the report is running.
- SYS\_REPORT\_NAME: Provides the name of the report name stored in repository.
- SYS\_USER\_PARAMS: Provides the string of user parameters with values.
- SYS\_SORT\_PARAMS: Provides the string of sort parameters.
- SYS\_REQUEST\_ID: A unique ID allotted by report server to the report.
- SYS\_PAGENO: Current Page no.
- SYS\_FILTER\_PARAMS: All filter parameters and values.
- SYS\_GROUPBY: All "group by" parameters in Adhoc reports.
- SYS\_USERID: The user ID using which user logged into Studio.
- SYS\_ORGID: The ORG ID of the user who has logged into Studio.

- 
- **SYS\_FIRST\_RECORD**: Returns -1 if current record is the first record in result-set. Returns 0 for other records. This is useful to check "First record" or "Not first record" conditions during scripting.
  - **SYS\_LAST\_RECORD**: Returns -1 if current record is the last record in result-set. Returns 0 for other records. This is useful to check "Last record" or "Not last record" conditions during scripting.

### **Parameter Input Dialog box in Studio: When will it appear**

Parameter input dialog box will appear in following cases:

1. When you open the report or refresh the fields (and its data source is stored procedure).
2. When you change the data-type of any parameter and click **OK** button of the **Parameter List** Dialog.
3. When you click the **Preview** Tab to preview the report.
4. When you click the **OK** Tab of **SQL Editor**.

Once parameter values are set, they are available till the report is open. A parameter's value is set to Null when its data type is changed.

Parameter input dialog box will not appear in the following cases:

If the data source is SQL then –

1. While opening a report having parameters.
2. Refreshing the fields of an open report having parameters added.
3. Verify SQL of Chart/Cross-tab.

