



GUI Programmers Guide 3.1



Jilroy Software LTD 2009

Jilroy Software makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Jilroy Software shall not be held liable for errors contained herein or direct, indirect, special, incidental, or consequential damages in connection with the furnishing, performance, or use of this material.

All rights are reserved. No part of this document may be photocopied, reproduced, or translated to another language without the prior written consent of Jilroy Software. The information contained in this document is subject to change without notice.

Copyright © 2000-2008 Jilroy Software. All rights reserved.

Reproduction, adaptation, or translation of this document without prior written permission is prohibited, except as allowed under the copyright laws.

Microsoft® is a US registered trademark of Microsoft Corporation.

Windows NT® is a US registered trademark of Microsoft Corporation.

Windows® 2000 is a US registered trademark of Microsoft Corporation.

Windows® and MS-Windows® are US registered trademarks of Microsoft Corporation.

Pentium® is a US registered trademark of Intel Corporation.

UNIX® is a registered trademark of The Open Group.

All other company and product names may be trademarks or registered trademarks of their respective owners.

Table of Contents

Preface.....	6
Introduction.....	8
The GUI structure overview.....	9
The query panel structure.....	10
The Menu Section.....	11
The Panel Section.....	11
The customized panels sections.....	11
The Swing Menu Configuration.....	12
The Menu XML format.....	12
isRoot.....	13
Name.....	13
Item.....	13
Item XML format.....	13
Type.....	14
Name.....	14
Shortcut.....	14
Icon.....	14
FormName.....	14
Parameters tag.....	15
Parameter tag.....	15
Key tag.....	15
Value tag.....	15
Special 'Run Inventory' command.....	15
Menu Predefined Entries.....	17
Exit.....	17
About.....	17
Edit Local Config File.....	17
Edit Config File.....	17
Edit Provisioning File.....	17
Local License Infob.....	17
License Info.....	17
The HTML Menu Configuration.....	18
Menu.ini	19
Label.....	20
Entry types.....	20
Menu entry.....	20
Report entry.....	20
Graph entry.....	20
ShowPopup entry.....	21

Generating the HTML menu.....	21
Panels Configuration.....	23
InputForm XML format.....	23
FormName.....	23
FormType.....	24
InputFields.....	24
InputActions.....	24
InputField XML format.....	24
Name.....	25
Label.....	25
DefaultValue.....	26
DefaultValuePropFile.....	26
DefaultValuePropField.....	26
isNewLine.....	26
Size.....	26
Type.....	26
InputField Type Related fields.....	27
Time.....	27
TimeoffsetMin.....	27
FillWith.....	27
SQLQuery.....	28
SpecialQuery.....	28
The InputActions Section.....	28
InputActions XML format.....	28
InputActions.....	28
InputAction XML format.....	30
Name.....	30
Label.....	30
Type.....	31
isNewLine.....	31
Size.....	31
Icon.....	31
GoToForm.....	31
Parameters.....	31
Parameter.....	32
Key.....	32
Value.....	32
The QueryForm XML format.....	33
QueryForm XML format.....	33
FormName.....	33
FormType.....	33
QueryCommand.....	34
AutoRunQuery.....	34
AddCheckBoxToResults.....	34

QueryFields.....	34
QueryActions.....	34
QueryFields XML format.....	35
QueryFields.....	35
QueryField XML format.....	36
Condition.....	37
The QueryActions Section.....	37
QueryActions XML format.....	37
QueryActions.....	38
QueryAction XML format.....	38
Panels Predefined Actions.....	39
Common predefined actions.....	39
Refresh.....	39
Cancel.....	39
Save.....	39
Print to CSV.....	39
Select ALL.....	39
Reset Selection.....	39
Selected Record Details.....	39
Delete_Record_From_Table Details.....	40
Save_Record_In_Table Details.....	41
Add_Record_To_Table Details.....	42
Update_Record_In_Table Details.....	42
Inventory_on_selected_record.....	43
Job related actions.....	44
Kill selected.....	44
Delete selected.....	44
Restart selected.....	44
Server side SQL Queries definitions.....	46
The QueryQueries XML format.....	46
SQLQueries.....	46
SQLQuery XML format.....	47
Name.....	47
Command.....	47
Final Page.....	48
More Information.....	48
Contact Us.....	48
Sales.....	48
Product Management.....	48

Preface

Welcome to the GUI programmers guide 3.1. This chapter provides an introduction to the structure and assumptions of this guide.

The Purpose of This Guide

The Graphical User Interface of some of the Jilroy products was built in a way that will enable the users of the products to extend the product functionality in a simple way.

This guide contains information needed to develop and extend the GUI of some of the Jilroy swing applications.

Who Should Use This Guide

This guide is intended for programmers and the advanced users of the products who want to extend the supplied GUI.

Organization of This Guide

This guide is structured to reflect the following conceptual divisions:

- Preface – A description of the guide's purpose, intended audience, organization, and conventions.
- Introduction – A general description of Graphical User Interface and its extensions.
- Defining Menus. – A description of the how to define and extend the menu of the GUI.
- Defining Query panels and reports – A description on how to define new query panels.
- Final Page – Information about contacting Jilroy Software.

Conventions

The manual uses the following conventions:

- Names of dialog boxes, windows, and unnamed screen areas are displayed in *italics*.
- Names of buttons, tabs, check-boxes, and other screen elements are displayed in **bold**. For example, click **OK** or type the **Start date**.
- **This font** is used for text that you enter.
- `This font` is used for code, directory names, file names, and system activity.
- UPPERCASE is used for keys and acronyms.
- Steps that involve two or more selections from a menu may be presented as a combination of selections separated by an > angled bracket.

For example, when you see **File > New**, click the **File** menu on the menu bar. This will open a drop-down menu. Then select the **New** command.

- Cross-references are underlined. For example, see Chapter 2.
- Hyperlinks are underlined and blue.
- The ⓘ symbol signifies notes, which are used to provide extra or special information regarding the preceding topic.
- The *Italic* font style is used to *emphasize* words and phrases in special cases.

Introduction

The GUI (Graphical User Interface) of most of the products developed by Jilroy Software, as the products themselves was developed in a way that enables the user of the GUI, to tailor it to its exact needs.

The idea was to create a community of users for the products, that will extend them using the core services supplied by the products.

It was planned that as the products evolve, most of the extensions will be reviewed by Jilroy Software and if they make sense, then they will be added to the product core.

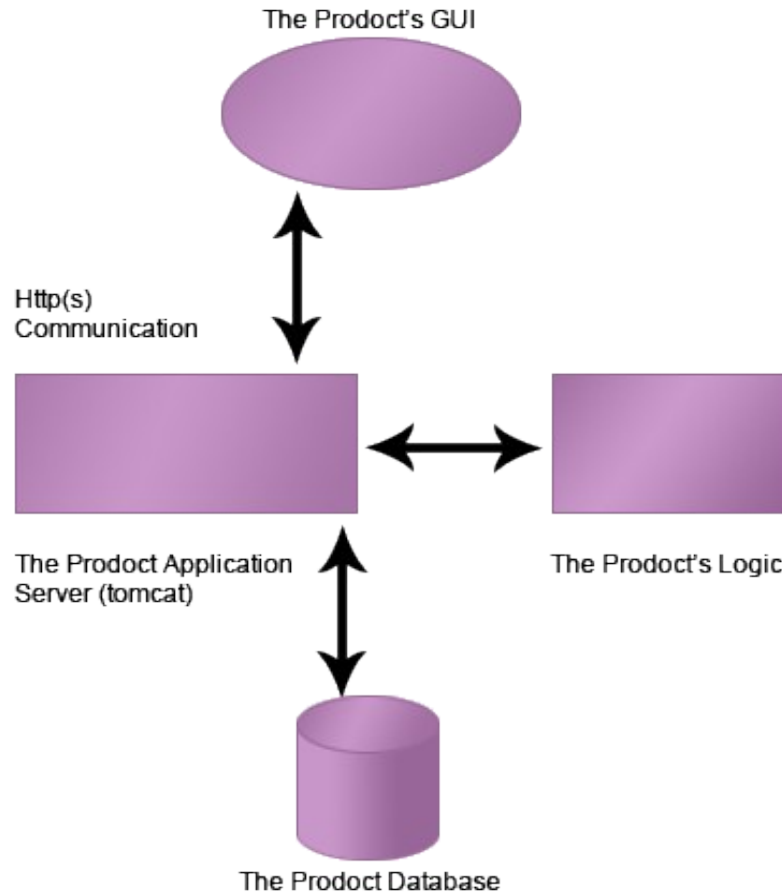
This guide will explain the general extensions that can be made to the GUI. Additional specific extensions will be explained in the specific products users guide.

Jilroy Software encourages you to use the product and extend it.

If you have any questions and recommendations about how to use or how to improve the product you are most welcome to send us your thoughts to support@jilroy.com

If you need us to tailor the product to your exact needs we are offering this kind of service for reasonable fees.

The GUI structure overview



There are 2 types of GUI supplied by the product:

- A Swing Full GUI
- An HTML reporter

The Swing GUI is a client written fully in swing. It contains only presentation logic, and all the application logic is performed in the product's application server which is a tomcat server.

The HTML reporter is a web browser based application which allows to view query reports as defined in the Swing GUI, also using a regular Internet browser such as Internet Explorer and Firefox.

The tomcat server communicates with the application's database, and other related files and programs.

The query panel structure

The Query Panel structure

The Menu Section

Administration Run Reports Help

Report Name: Nodes

The Query Fields Section


Query Fields:

IP Address: DNS Name: Node SysObjectId:

Discovery Method: Monitoring Method:

The Query Actions Section

Actions:

 Lines Limit:

The Query Results Section

Query Results:

NODE_OBJECTID	NODE_IP	DNS_NAME	SYSDESCRIPTION	SYSOBJECTID
2	10.0.1.203	null	null	null
4	10.0.1.201	null	null	null
6	10.0.1.254	null	Cisco Internetwork Ope...	1.3.6.1.4.1.9.1.615
57	10.0.1.202	null	null	null

A common GUI panel in the Swing & the HTML form will look like the capture shown above.

We can identify 2 major sections in the panel

- The Menu section
- The Panel Body

Both these sections can be customized by the users, using configuration files.

The Menu Section

The menu section represents the main menu of the application. It can contain sub-menus and action elements

The Panel Section

The panel section, contains the relevant data selected by the user. There are pre-made panels, which can be used as-is by the users, and there are the customized panels like the one shown above which is fully customized by the user.

The customized panels sections

The customized panel is divided into 3 sections:

- The query fields section
- The query actions section
- The query results section

All these sections can be customized by the user by configuring XML files, that are put in a specific directory, as will be explained later.

The Swing Menu Configuration

The GUI has a main menu, which can be fully customized by the user. The customization is done editing XML files which are located in the **[INSTALLDIR]/conf/menus** directory.

There could be as many configuration files as wanted in this directory and there is no restriction to their name, and the relations they define.

Each menu has its own file which holds its definition. Menus are pointed by other menus to create the menu tree.

The Menu XML format

The following section will list the tags available in the menu definition.

```
<Menu>
  <IsRoot>1</IsRoot>
  <Name>TopMenu</Name>
  <Item>
    <Name>Administration</Name>
    <Type>Menu</Type>
  </Item>
  ...
  <Item>
    <Name>Discovery Jobs</Name>
    ...
  </Item>
</Menu>
```

isRoot

A Menu can be a root menu, or a sub menu. This key defines which type is this. It contains only one type of child which is the Menu tag.

The name of the Top Menu selected is specified in the configGUI.xml of the product's GUI under the tag <TopMenuName>

Valid values are:0, 1

Name

The name of the Menu entry. This name should be unique and it is recommended that it will be descriptive. This field is sometimes used by other Menu items for reference.

Valid values: a string, not tampering with the XML format.

Item

The Item XML defines the lines that will be within the menu definition. Following is a detailed explanation of the structure of an Item XML.

There could be any number of Item entries within the Menu definition.

Item XML format

The following section will list the tags available in the menu definition.

```
<Item>
    <Name>Administration</Name>
    <Type>Menu | Command | SqlQueryPanel</Type>
    <Shortcut>U</Shortcut>
    <Icon>icon path</Icon>
    <FormName>EditPropertiesFile</FormName>
    <Parameters>
        <Parameter>
            <Key>PropertiesFilePath</Key>
            <Value>
```

```
    /opt/jilroy/rjnetworkdiscovery/conf/connection.prop
    </Value>
  </Parameter>
</Parameters>
</Item>
```

Type

The type of the menu item, it can be:

- **Menu** – In this case it points to a sub-menu whose name is given in the Name tag.
- **InputPanel** – In this case the Name tag points to a panel definition, as explained in the coming sections.
- **SqlQueryPanel** – In this case the Name tag points to a panel definition, as explained in the coming sections.
- **Command** – A command. In this case the value of the tag 'Name' is passed to the menu action listener (This is used for predefined commands).
- **Separator** – This tag adds a separator to the menu, at the given location

Name

The name of the entry of the type specified in the Name tag, as explained there. There are a few predefined entries, which can be used by the user.

Shortcut

The shortcut value for selecting the item.

Icon

The Icon that will appear near the text of the menu item.

FormName

This is an optional tag, used with special generic forms supplied by the product. It overrides the Name Value in the search of a panel definition.

Currently we have the following generic pre-made forms:

- PropertiesFilePath

This form displays the fields in a local properties file (in the format of conf/config.conf)

It accepts the following parameter:

- PropertiesFilePath – and its value is the full path to the properties file.

Parameters tag

This is an optional tag, used with special generic forms supplied by the product. It contains parameters for these forms.

It can hold any number of parameters.

Its structure is:

```
<Parameters>
  <Parameter>
    <Key>name</Key>
    <Value>value</Value>
  </Parameter>
</Parameters>
```

Parameter tag

This Tag holds the key and value tags of the parameter.

Key tag

This Tag holds the key name of the parameter.

Value tag

This Tag holds the value of the key found near it.

Special 'Run Inventory' command

There exists a special predefined command named 'Run Inventory'. This command will launch an “inventory” job based on the parameters passed to the command.

Following is an example of the parameters that define which Inventory job will be executed.

```
<Parameters>
  <Parameter>
    <Key>Command</Key>
    <Value>Run Inventory</Value>
  </Parameter>
  <Parameter>
    <Key>DiscoveryItem</Key>
    <Value>Layer2_all_router_mapping</Value>
  </Parameter>
  <Parameter>
    <Key>ParametersFile</Key>
    <Value></Value>
  </Parameter>
</Parameters>
```

The parameters meaning is:

- Command – specifies that this is the predefined 'Run Inventory' command.
- DiscoveryItem – defines the name of the discovery item which will be discovered
- ParametersFile – an optional parameter that specifies the name of an .ini file containing parameters that should be passed to the Inventory job.

Menu Predefined Entries

This chapter lists the predefined entries in the menu definitions

Exit

Causes the program to end.

About

Causes the program to show in a message box, the contents of the conf/version.txt file

Edit Local Config File

Causes the program to show the contents of the file pointed by the <IniFile> tag found in the config.xml.

Edit Config File

Causes the program to request the application server to bring its own conf/config.prop file.

Edit Provisioning File

Causes the program to request the application server to bring its own conf/config.prop file.

Local License Infob

Causes the program to show the information of the local license file located in the conf/ dir.

License Info

Causes the program to request the application server to show the contents of its license information.

The HTML Menu Configuration

As the HTML GUI has limited capabilities compared to the ones of Swing GUI, the HTML menu has also limited capabilities compared to the ones of the Swing GUI.


You as the developer of the HTML GUI can control only the sub menus of the top menu entries: Reports & Graphs, as can be seen in the following image

The screenshot displays the Jilroy Software LTD ReportingSystem interface. The top navigation bar includes the Jilroy logo, the text 'SOFTWARE LTD', and two menu items: 'Reports' and 'Graphs'. Below the navigation bar is an orange banner with the text 'Monitor Your Enterprise Your Way' and a stylized graphic. The main content area is titled 'Report Type: Interfaces' and contains three sections: 'Query Fields', 'Actions', and 'Results'.

Query Fields

IP Address: Interface Alias: Interface Description:

Actions

 Line Limits:

Results

IF_OBJECTID	MODE_IP	IF_INDEX	IF_DESCRIPTION	IF_NAME	IF_ALIAS
18	10.0.1.254	3	Vlan3	null	null
19	10.0.1.254	10108	GigabitEthernet0/8	null	null
20	10.0.1.254	10122	GigabitEthernet0/22	null	null
21	10.0.1.254	10111	GigabitEthernet0/11	null	null
22	10.0.1.254	5011	Port-channel11	null	null
23	10.0.1.254	10105	GigabitEthernet0/5	null	null

You do not need any knowledge in HTML or JavaScript in order to control the menu format. You have to update an INI file containing the menu format and then run a command that will automatically generate the HTML files needed to operate the menu.

Menu.ini

This file is located under [COMPANYDIR]/tomcat/jilroy/ path. It contains a sample definition of the default HTML menu.

```
[Reports]
Discovery=Menu:Discovery_Reports
Monitoring=Menu:Monitoring_Reports
[Graphs]
Discovery=Menu:Discovery_Graphs
Monitoring=Menu:Monitoring_Graphs
[Discovery_Reports]
Nodes=Report:NodesReport,Nodes
Interfaces=InterfacesReport,Interfaces
[Discovery_Graphs]
[Monitoring_Reports]
Interfaces status=Report:InterfacesStatusReport,Interfaces Status
Service level=Report:ServiceLevelReport,Service Level Report
[Monitoring_Graphs]
RRD Graphs=RRD:RRDGraph
```

The entries in [] are the names of the sub-menus when [Reports] & [Graphs] are also the top menu entries (they appear at the top level of the html menu)..

Under each sub-menu entry you will enter the sub-menu items which are identified in the format of:

label=Entry_type:Parameters

Label

The label is the value that will be displayed in the sub-menu when it is displayed.

Entry types

Currently these are the types of menu items supported:

Menu entry

This type specifies that pointing on this entry in the menu will open a new sub menu.

There must be a sub-menu entry with this name.

The parameter of this type key is the name of the sub-menu entry.

Report entry

This type specifies that selecting this entry in the menu will display the selected report

The parameters of this type are:

1. The name of the swing panel describing the report
2. The name of the SQL query that should generate the output

Graph entry

This type specifies that selecting this entry in the menu will display the selected graph

The parameters of this type are:

1. The name of the swing panel describing the graph
2. The name of the Graph definition that should generate the output

ShowPopup entry

This type specifies that selecting this entry in the menu will display a popup with the first line of query results as passed in the parameters of this entry.

An example of an such an entry is:

```
<Item>
  <Name>Properties...</Name>
  <Type>ShowPopup</Type>
  <Parameters>
    <Parameter>
      <Key>QueryName</Key>
      <Value>SampleQuery</Value>
    </Parameter>
  </Parameters>
</Item>
```

The parameters passed should have a key named QueryName, and the value is the name of the query as defined in one of the SQL queries defined in the conf/sqlqueries directory.

This option is commonly used in the popup menus of map elements.

The SQL query selected is appended with the “userdata” value of the selected icon, and in this way can be used as a dynamic query related to the discovered node

Generating the HTML menu

The Menu.ini defines the format of the HTML menu used. Converting it to HTML is done using a utility named: **generate_js_menu**

Usage: generate_js_menu Role_name Menu.ini

Parameters:

- Role_name: The role for it the menu will be displayed. This is the way the security mechanism is implemented.
- Menu.ini: The path to the Menu.ini file.

The output of the utility is a .js file containing the menu definitions, suitable for all types of browsers.

Panels Configuration

The user can customize some of the panels that will be displayed using the GUI.

The customization is done editing XML files which are located in the **[INSTALLDIR]\conf\panels** directory.

There could be as many configuration files as wanted in this directory and there is no restriction to their name, usually each file defines a single panel.

The panels are pointed by the MenuItem definitions.

InputForm XML format

The following section will list the tags available in the InputForm definition.

```
<InputForm>
  <FormName>form name</FormName>
  <FormType>InputPanel</FormType>
  <InputFields>
    ...
  </InputFields>
  <InputActions>
    ...
  </InputActions>
</InputForm>
```

FormName

The form name. This is the name specified in the MenuItem Name definitions in order to point to the given Panel.

FormType

The form type. Currently there is only 1 type supported:

- InputPanel – for Simple input panels

InputFields

This is an XML tag that defines the parameters that the user will be able to set in the panel. We will explain in the following sections, the exact structure of this field.

InputActions

This is an XML tag that defines the actions that can be performed by the user on this panel.

There are a set of predefined actions that you can select. We will explain in the following sections, the exact structure of this field.

InputField XML format

The following section will list the tags available in the InputField definition.

```
<InputField>
  <Name>strDescription</Name>
  <Label>Job Description:</Label>
  <Size>100</Size>
  <Type>String</Type>
  <DefaultValue>1000</DefaultValue>
</InputField>
<InputField>
  <Name>UploadUpoorted</Name>
  <Label>Upload Supported:</Label>
  <Size>30</Size>
  <Type>CheckBox</Type>
  <DefaultValuePropFile>location/conf/config.prop</DefaultValuePropFile>
  <DefaultValuePropField>Client.Upload_supported</DefaultValuePropField>
```



```
</InputField>
<InputField>
  <Name>GraphType</Name>
  <Label>Graph Type:</Label>
  <Size>100</Size>
  <Type>Combo</Type>
  <FillWith>
    <SpecialQuery>GetRRDDbTypes</SpecialQuery>
    <SQLQuery></SQLQuery>
  </FillWith>
</InputField>
<InputField>
  <Name>StartTime</Name>
  <Label>Start Time:</Label>
  <isNewLine>1</isNewLine>
  <Size>30</Size>
  <Type>DateTimePicker</Type>
  <Time>
    <TimeOffsetMin>-10</TimeOffsetMin>
  </Time>
</InputField>
```

There are multiple types of InputField entries, each with some special parameters.

Name

The Name tag, is the name of the field we are going to add to the SQL query condition.

Label

The Label tag, is the display name that appears near the input field, on the screen of the GUI.

DefaultValue

This tag is an optional tag which defines a default value of the given field. If no value is specified, than a NULL value is assumed.

DefaultValuePropFile

This tag is an optional tag which defines the source of the DefaultValue when it is taken from a prop file. The tag points to the location of the file.

DefaultValuePropField

This tag is an optional tag which defines the tag that contains the default value of the field. It overrides the DefaultValue settings.

isNewLine

This tag affects the layout of the display. It defines whether the current FieldItem will appear in the first position of a new line, or it will continue the previous items line.

Size

This tag affects the layout of the display. It defines the minimum size of the input field of this QueryField.

Type

This tag determines the type of the query field. The currently supported types are:

- String – A textbox query field
- Combo – A combo box selection, where the user has to choose from a list.
- DateTimePicker – A simple date and Time picker field group.
- FileChooser – A textbox with a file chooser icon near it
 - Supports an additional attribute <InitialPath> for file chooser initial directory path.
- DirectoryChooser – A textbox with a directory chooser icon near it
 - Supports an additional attribute <InitialPath> for file chooser initial directory path.
- FileDirectoryChooser – A textbox with a file or directory chooser icon near it

- Supports an additional attribute <InitialPath> for file chooser initial directory path.
- CheckBox – a CheckBox field that returns a value of (0|1)

InputField Type Related fields

Time

The Time tag is relevant with the DateTimePicker type. It enables the user to set the time in the Window.

```
<Time>  
    <TimeOffsetMin>-10</TimeOffsetMin>  
</Time>
```

TimeoffsetMin

The TimeOffsetMin tag is used to allow the user to set the time, in the calander. The time is relative to now, and can be positive or negative. It specifies minutes from now.

FillWith

The FillWith tag is relevant with the Combo type. It enables the user to fill the combo box.

```
<FillWith>  
    <SpecialQuery>GetRRDDbTypes</SpecialQuery>  
    <SQLQuery>query name</SQLQuery>  
</FillWith>
```

SQLQuery

An SQL query name, that will have data that will fill the combo. Only the first field selected is entered.

SpecialQuery

The SpecialQuery tag defines special queries that can return a list of selections. Currently supported special queries are:

- GetRRDDbTypes – return a list of existing RRD report types.
- GetRRDDbList – return a list of existing RRD databases existing under a given report. It must appear in the same panel with the GetRRDDbTypes

The InputActions Section.

This is an XML tag that defines the actions that can be performed by the user on this panel.

There are a set of predefined actions that you can select.

InputActions XML format

The following section will list the tags available in the QueryActions definition.

```
<InputActions>
  <InputAction>
    ...
  </InputAction>
  ...
  <InputAction>
    ...
  </InputAction>
</InputActions>
```

InputActions

This is a wrapping tag for all InputAction definitions.

It contains only one type of sub XML which is a InputAction. There could be multiple entries of InputAction XML in it.

InputAction XML format

The following section will list the tags available in the InputAction definition.

```
<InputAction>
    <Name>Refresh</Name>
    <Type>Button|InputField</Type>
    <Label>Refresh</Label>
    <Size>100</Size>
    <isNewLine>1</isNewLine>
    <GoToForm>ResultSetDetailsForm</GoToForm>
    <Icon>/opt/jilroy/rjmonitoringplatform//resource/refresh.jpeg</Icon>
    <Parameters>
        <Parameter>
            <Key>Command</Key>
            <Value>Delete From Permissions Table</Value>
        </Parameter>
    </Parameters>
</QueryAction>
```

Name

This is the Name for the action which is associated with the action. Usually this can not be changed.

There are some predefined names that define known actions. look at the following section for a list those actions.

Label

This is the value that appears on the button if no icon is supplied. Otherwise this is the tooltip for the button.

Type

This tag defines the action type. The currently supported types are:

- Button – The action format is of a button.
- InputField – The action field is an input field which has exactly the same tags as the QueryField XML tag.

isNewLine

This tag affects the layout of the display. It defines whether the current FieldAction will appear in the first position of a new line, or it will continue the previous items line.

Size

This tag defines the minimum size of the button, if no Icon is supplied.

Icon

This tag points to the button's Icon.

GoToForm

This tag is used for actions that leads to other forms. This tag points to the panel which we want to go to.

Parameters

This is an optional tag, used with special predefined actions.
It contains parameters for these forms.
It can hold any number of parameters.

Its structure is:

```
<Parameters>
  <Parameter>
    <Key>name</Key>
    <Value>value</Value>
```

```
</Parameter>  
</Parameters>
```

Parameter

This Tag holds the key and value tags of the parameter.

Key

This Tag holds the key name of the parameter.

Value

This Tag holds the value of the key found near it.

The QueryForm XML format

QueryForm XML format

The following section will list the tags available in the QueryForm definition.

```
<QueryForm>
  <FormName>All Jobs Status</FormName>
  <FormType>Report</FormType>
  <QueryCommand>All Jobs Status</QueryCommand>
  <AutoRunQuery>1</AutoRunQuery>
  <AddCheckBoxToResults>1</AddCheckBoxToResults>
  <QueryFields>
    ...
  </QueryFields>
  <QueryActions>
    ...
  </QueryActions>
</QueryForm>
```

FormName

The form name. This is the name specified in the MenuItem Name definitions in order to point to the given Panel.

FormType

The form type. Currently there are 2 types supported:

- Report – for SQL query based reports
- Graph – for graph based reports.

QueryCommand

For Type = Report, this parameter specifies an SQL query entry name, defined in the Application server. We will explain later how to define the SQL Commands entries.

The form name. This is the name specified in the MenuItem Name definitions in order to point to the given Panel.

AutoRunQuery

This Tag indicates if the query selected should be executed (i.e the refresh key, should be simulated when the panel is selected) or not.

Valid values are:0, 1

AddCheckBoxToResults

Some times it is needed to perform actions on the some selected rows in the result set (look at the job status reports). This Tag adds a check box to the end of every line.

QueryFields

This is an XML tag that defines the parameters that the user will be able to set as filters for the query. They are added to the sql query in the conditions part. We will explain in the following sections, the exact structure of this field.

QueryActions

This is an XML tag that defines the actions that can be performed by the user on this panel.

There are a set of predefined actions that you can select. We will explain in the following sections, the exact structure of this field.

QueryFields XML format

The following section will list the tags available in the QueryFields definition.

```
<QueryFields>
  <QueryField>
    ...
  </QueryField>
  ...
  <QueryField>
  </QueryField>
</QueryFields>
```

QueryFields

This is a wrapping tag for all QueryField definitions.

It contains only one type of sub XML which is a QueryField. There could be multiple entries of QueryField XML in it..

QueryField XML format

The following section will list the tags available in the QueryField definition.

```
<QueryField>
  <Name>strDescription</Name>
  <Label>Job Description:</Label>
  <Size>100</Size>
  <Type>String</Type>
  <Condition>Like</Condition>
  <DefaultValue>1000</DefaultValue>
</QueryField>
<QueryField>
  <Name>GraphType</Name>
  <Label>Graph Type:</Label>
  <Size>100</Size>
  <Type>Combo</Type>
  <FillWith>
    <SpecialQuery>GetRRDDbTypes</SpecialQuery>
    <SQLQuery></SQLQuery>
  </FillWith>
</QueryField>
<QueryField>
  <Name>StartTime</Name>
  <Label>Start Time:</Label>
  <isNewLine>1</isNewLine>
  <Size>30</Size>
  <Type>DateTimePicker</Type>
  <Time>
    <TimeOffsetMin>-10</TimeOffsetMin>
  </Time>
</QueryField>
```

```
</QueryField>
```

There are multiple types of QueryField entries, each with some special parameters. The parameters which are common to the InputField are explained in the InputField section

Condition

This tag specifies what relation to use with the existing field condition. Valid values are:

- Like
- >=
- <=
- IS NOT

The QueryActions Section.

This is an XML tag that defines the actions that can be performed by the user on this panel.

There are a set of predefined actions that you can select.

QueryActions XML format

The following section will list the tags available in the QueryActions definition.

```
<QueryActions>
  <QueryAction>
    ...
  </QueryAction>
  ...
  <QueryAction>
    ...
  </QueryAction>
```

```
</QueryActions>
```

QueryActions

This is a wrapping tag for all QueryAction definitions.

It contains only one type of sub XML which is a QueryAction. There could be multiple entries of QueryAction XML in it.

QueryAction XML format

The following section will list the tags available in the QueryAction definition. It has the same parameters as the InputAction tag explained before. See the InputAction documentation for detailed information.

```
<QueryAction>
    <Name>Refresh</Name>
    <Type>Button|InputField</Type>
    <Label>Refresh</Label>
    <Size>100</Size>
    <isNewLine>1</isNewLine>
    <Icon>/opt/jilroy/rjmonitoringplatform//resource/refresh.jpeg</Icon>
    <GoToForm>ResultSetDetailsForm</GoToForm>
    <Parameters>
        <Parameter>
            <Key>Command</Key>
            <Value>Delete From Permissions Table</Value>
        </Parameter>
    </Parameters>
</QueryAction>
```

Panels Predefined Actions

This chapter lists the predefined actions names and their use.

Common predefined actions

Refresh

This commands refresh the query done in an SQL Query pannel

Cancel

This command resets the panel showed and returns to the main frame.

Save

With a PropertiesFile Type of Panel, it saves the File locally.

Print to CSV

With a Query Panel, it allows the user to save the output table content in a CSV file, for later printing with a spreadsheet product.

Select ALL

With a Query Panel that has a selectable checkbox in its end, it allows to select all raws.

Reset Selection

With a Query Panel that has a selectable checkbox in its end, it allows to un-select all raws.

Selected Record Details

With a Query panel that has a selectable checkbox in its end, it allows to view a

detailed screen about the first selected row.

It uses several parameters to identify the requested record.

```
<GoToForm>ResultSetDetailsForm</GoToForm>
<Parameters>
  <Parameter>
    <Key>Query</Key>
    <Value>NodesDetailesSQLCommand</Value>
  </Parameter>
  <Parameter>
    <Key>KeyFieldNickName</Key>
    <Value>NODE_OBJECTID</Value>
  </Parameter>
  <Parameter>
    <Key>KeyFieldTrueName</Key>
    <Value>nObjectId</Value>
  </Parameter>
  <Parameter>
    <Key>KeyFieldType</Key>
    <Value>Integer</Value>
  </Parameter>
</Parameters>
```

- GoToForm - points to the ResultSetDetailsForm which lists the output of a query in a vertical way, when each field is in a line of its own.
- The Query parameter points to the SQL query used by the "ResultSetDetailsForm" form.
- The " KeyFieldNickName" specifies to the nickname of the key field in the record selected
- The " KeyFieldTrueName" specifies to the true table name of the key field in the record selected
- The "KeyFieldType" specifies the type of the field.

Delete_Record_From_Table Details

With a Query panel that has a selectable checkbox in its end, it allows to delete records from tables based on the selected record.

It uses several parameters to identify the records to delete. Each set of Parameters define a set of records to delete. Many <Parameters> entries are

allowed.

```

    <Parameters>
      <Parameter>
        <Key>SQLQuery</Key>
        <Value>DeleteSwitch</Value>
      </Parameter>
      <Parameter>
        <Key>DeleteKeyField</Key>
        <Value>STRIPADDR</Value>
      </Parameter>
      <Parameter>
        <Key>DeleteQueryKey</Key>
        <Value>SWITCH_IP</Value>
      </Parameter>
    </Parameters>
    <Parameters>
      <Parameter>
        <Key>SQLQuery</Key>
        <Value>DeleteSwitchInterfaces</Value>
      </Parameter>
      <Parameter>
        <Key>DeleteKeyField</Key>
        <Value>strSwitchIpAddr</Value>
      </Parameter>
      <Parameter>
        <Key>DeleteQueryKey</Key>
        <Value>SWITCH_IP</Value>
      </Parameter>
    </Parameters>

```

- SQLQuery - defines the name of the entry in an .sql file that contains the delete to be executed. This command can contain conditions inside it.
- DeleteKeyField - This is the field we will use in the “where” clause to identify which record to delete
- The “DeleteQueryKey” specifies the name of the field (as appears in the form) from which to take the value of the “DeleteKeyField”.

Save_Record_In_Table_Details

In a panel that displays the content of a record and allows modifying it, this command specifies where we want to save this record.

It uses several parameters to identify the table where to insert or update the table.

```
<Parameters>
  <Parameter>
    <Key>SQLQuery</Key>
    <Value>Layer2SwitchUniqueTest</Value>
  </Parameter>
  <Parameter>
    <Key>SQLQueryTableName</Key>
    <Value>Layer2SwitchQueryTableName</Value>
  </Parameter>
  <Parameter>
    <Key>UniqueFields</Key>
    <Value>strIpAddr,nSnmpGetPort</Value>
  </Parameter>
</Parameters>
```

- SQLQuery - defines the name of the entry in an .sql file that contains the select command to identify if the record already exists, and then an update will be done, or otherwise an insert will be done.
- SQLQueryTableName - defines the name of the entry in an .sql file that contains the name of the table to insert to, or update.
- The " UniqueFields" specifies the name of the fields that together from the "where" clause that identifies if the record already exists.

Add_Record_To_Table Details

This command is usually used to open a new screen that contains modifiable fields, and clears those fields.

Update_Record_In_Table Details

In a query panel that displays a list of records, selecting a record and pressing a button with this command will usually lead to a panel with modifiable fields, that will contain the details of the selected record.

It uses several parameters to identify the selected record keys.

```
<Parameters>
  <Parameter>
    <Key>SQLQuery</Key>
    <Value>Select For update Layer2 Switch</Value>
  </Parameter>
  <Parameter>
    <Key>KeyField</Key>
    <Value>STRIPADDR</Value>
  </Parameter>
  <Parameter>
    <Key>QueryKey</Key>
    <Value>SWITCH_IP</Value>
  </Parameter>
</Parameters>
```

- SQLQuery - defines the name of the entry in an .sql file that contains the select command to identify the record we want to update
- KeyField - Identifies the key field for the select query
- QueryKey - Identifies the column name in the panel containing the key field's value.

Inventory_on_selected_record

In a query panel that displays a list of records, selecting a record and pressing a button with this command will launch an Inventory job with the parameters identifying what we want to discover.

It uses several parameters to select the parameters for the discovery process.

```
<Parameters>
  <Parameter>
    <Key>SQLQuery</Key>
    <Value>Layer2SwitchesInventoryCommand</Value>
  </Parameter>
  <Parameter>
    <Key>UniqueFields</Key>
    <Value>stripAddr</Value>
  </Parameter>
  <Parameter>
    <Key>QueryFields</Key>
    <Value>SWITCH_IP</Value>
  </Parameter>
</Parameters>
```

```
</Parameter>
<Parameter>
  <Key>SQLQueryTableName</Key>
  <Value>Layer2SwitchQueryTableName</Value>
</Parameter>
<Parameter>
  <Key>Discovery_rule</Key>
  <Value>Layer2_switch_mapping</Value>
</Parameter>
</Parameters>
```

- SQLQuery - defines the query to select to supply the record which will be the parameters list to the inventory command.
- UniqueFields - a CSV identifying the unique fields Identifies the key field for the select query
- SQLQueryTableName - the name of the table on which the SQL query will run.
- Discovery_rule – the name of the discovery rule that will be launched.

Job related actions

The following list of commands are job related predefined commands

Kill selected

This command kills the selected jobs in the jobs query panel.

Delete selected

This command deletes the selected jobs in the jobs query panel.

Restart selected

This command restarts the selected jobs in the jobs query panel.

Server side SQL Queries definitions

We did not want to define queries on the client side for security reasons. That is why the reference to the SQL commands is by name.

The SQL commands themselves are defined in the server side, in the directory [INSTALLDIR]/conf/sqlqueries

Currently you can find references to the SQL commands in the QueryForm definitions.

The QueryQueries XML format

The following section will list the tags available in the QueryForms definition.

```
<SQLQueries>
  <SQLQuery>
    ...
  </SQLQuery>
  ...
  <SQLQuery>
    ...
  </SQLQuery>
</SQLQueries>
```

SQLQueries

This is a wrapping tag for all SQLQuery entries defined in a given file.. It should be the top tag in every query form definitions file.

It contains only one type of sub XML which is a SQLQuery. There could be any number of SQLQuery enties within a given file.

SQLQuery XML format

The following section will list the tags available in the SQLQuery definition.

```
<SQLQuery>
  <Name>Nodes</Name>
  <Command>
    SELECT
      DISTINCT Nodes.NOBJECTID Node_ObjectId,
      Nodes.STRIPADDR Node_IP,
      Nodes.STRDNSNAME DNS_Name,
      Nodes.STRDESCRIPTION SysDescription,
      Nodes.STRSYSOBJECTID SysObjectId
    FROM
      TBLOBJECT Nodes,
      TBLOBJECT Addresses
    WHERE
      Nodes.strType = 'Node' AND
      Addresses.strType = 'Address' AND
      Nodes.nObjectId = Addresses.nParentObjectId
  </Command>
</SQLQuery>
```

Name

This tag determines the name of the entry. It is used for reference by other GUI side elements.

Command

This tag contains the actual sql command.

On its end the relevant conditions are appended.

Final Page

More Information

More information about Jilroy Software, the GUI programmers reference 3.1, and our other products can be found on our web site.

www.jilroy.com

Contact Us

For any information or problem, request for information or extension idea related to our GUI or any other product, please contact one of the following email addresses.

Sales

sales@jilroy.com

Product Management

[discovery pm@jilroy.com](mailto:discovery_pm@jilroy.com)

FTP Site

All jilroy products can be downloaded from our Web site.