

Customizing Getting Started Content

The Getting Started tab is displayed as a user enters Exago. This tab can be customized by loading custom html. This is done by modifying the language element 'GettingStartedContent' in the file 'en-us-getting-started.xml'. To assist in customizing the Getting Started tab, Exago provides several JavaScript functions to open the New Report Wizard, execute reports, open other custom tabs and display reports as dashboards.

The example below demonstrates a custom tab with links to the New Report Wizard and Dashboards.

Revenue by Category:

Category	Percentage
Produce	30.06%
Meat/Poultry	26.93%
Seafood	5.14%
Grains/Cereals	2.07%
Dairy Products	16.83%
Confections	6.14%
Beverages	11.62%
Condiments	1.20%

Number of Sales by Employee:

Employee	Number of Sales
Buchanan, Steven	117
Callahan, Laura	260
Davolio, Nancy	343
Dodsworth, Anne	104
Fuller, Andrew	239
King, Robert	176
Leverling, Janet	319
Peacock, Margaret	420
Suyama, Michael	168
Total Number of Sales	2,146.00

Note: It is recommended to make custom tabs in a separate language file to make it easy to change tabs by user or groups of users. See **Modifying Select Language Elements**.

Creating Additional Custom Tabs

Additional custom tabs can be created by creating two language elements with unique names. One element specifies the title of the custom tab and the second contains the html content. Custom tabs can be opened with the JavaScript function `wrAddTabbedContent` (see **Available JavaScript Functions**).

The example below demonstrates a custom tab that has buttons to launch reports.

```
<element id="QuickReportsTabName">Quick Reports</element>
<element id="QuickReportsTab">
  <style type="text/css">
    .Button
```

```

    {
        height:20px;
        width: 60px;
        color: black;
        font-size:8pt;
        margin-right:5px;
    }
    .divProductDescription
    {
        margin-bottom:3px;
    }
</style>
<p style="font-family:Arial; font-size:12pt; font-weight:bold; text-
decoration:underline; text-align:center; margin-bottom:10px;">Click the format below the
report you want to run. </p>
<br />
<div class="divProductDescription">
    <b>Revenue by Category (with drilldown)</b> - Complete list of revenue
generated by each category of products.
</div>
<div class="divProductButtons">
    <input type="button" class="Button" value="HTML"
onclick="wrExecuteReport('Sales Reports\\Revenue by Category','html')" />
    <input type="button" class="Button" value="EXCEL"
onclick="wrExecuteReport('Sales Reports\\Revenue by Category','excel')" />
    <input type="button" class="Button" value="PDF"
onclick="wrExecuteReport('Sales Reports\\Revenue by Category','pdf')" />
    <input type="button" class="Button" value="RTF"
onclick="wrExecuteReport('Sales Reports\\Revenue by Category','rtf')" />
    <input type="button" class="Button" value="CSV"
onclick="wrExecuteReport('Sales Reports\\Revenue by Category','csv')" />
</div>
</element>

```

Available JavaScript Functions

To assist with the creation of custom tab content, Exago provides a small number of JavaScript functions to allow custom html to call features of Exago.

`void wrStartNewReportWizard()`

Description	Opens the New Report wizard in a new tab.
Example	Ex. <i>Click <code>here</code> to create a new report.</i>

`void wrStartDuplicateReportDialog(string reportFolder\\reportName):`

Description	Opens the Duplicate Report dialog.
--------------------	------------------------------------

Remark	If the report name is null or blank Exago will use the report selected in the Main Menu.
Example	Ex. <i>Click <code>here</code> to create a duplicate this report.</i>

`void wrExecuteReport(string reportFolder\\reportName, string format)`

Description	Executes the specified report in the specified format.
Example	Ex. <code><input type="button" class="Button" value="HTML" onclick="wrExecuteReport('Sales Reports\\Revenue by Category','html')></code>

`string wrGetSelectedReportName()`

Description	Returns the name of the report that is selected in the Main Menu.
Remark	The returned string will include the folder structure of the report separated by slashes.

`void wrAddTabbedContent(string ContentID, string TabName)`

Description	Opens a new tab and loads the html stored in the element of the Language file that corresponds to the Content ID.
Remark	The ContentID should match the element ID of the html you want to load. The TabName should make the element ID of the name you want the tab to display.

`data-onloadreportname= "ReportFolder\\ReportName"`

Description	Executes a report as HTML and loads it into a div or iframe.
Remark	The report string should be formatted as Report Folder \\ Report Name. Note: When using this function make sure the setting Enable Debugging in Other settings is False .
Example	Ex. <code><div class="Report" data-onloadreportname="Employee Reports\\Number of Sales by Employee"></div></code>

`data-useviewer = "True/False"`

Description	Specifies to load a report as raw html or utilize Exago dynamic report viewer.
Remark	Default value is True. In cases where the dynamic capabilities of the Exago viewer is not need set to False to load raw html.
Example	Ex. <code><div class="Report" data-onloadreportname="Employee Reports\Number of Sales by Employee" data-useviewer="False"></div></code>

data-enablescrolling = "True/False"

Description	Specifies whether or not to show scroll bars.
Remark	Default value is True. This can helpful for certain reports that may not fit exactly within the startup content.
Example	Ex. <code><div class="Report" data-onloadreportname="Employee Reports\Number of Sales by Employee" data-enablescrolling="False"></div></code>

data-reloadinterval="n"

Description	Reloads a report every <i>n</i> seconds.
Remark	This function is used in conjunction with data-onloadreportname .
Example	Ex. <code><div class="Report" data-onloadreportname="Employee Reports\Number of Sales by Employee" data-reloadinterval="2"></div></code>

data-allowexport="0/1"

Description	Specifies wether or not to show the re-export menu for the report.
Remark	The default value is 0 (does not show the menu). Set to 1 to have the re-export options display.
Example	Ex. <code><div class="Report" data-onloadreportname="Employee Reports\Number of Sales by Employee" data-reloadinterval="1"></div></code>