

Configure the Flowtime Website

Last Modified on 08/08/2022 6:10 am EDT

V9.2

You configure the Flowtime site to provide users with the smooth functionality and usability. You can customize the pages to comply with your organization's branding policies, add or edit existing pages, set up functionality that fit existing processes, and configure new menus and submenus.

Configuration files

After you install the Flowtime site, you can configure the following files:

File type	Description	Location
web.config	Controls settings, such as culture, theme, and database connection. For example, you can set the Flowtime website language by editing the following key: <code><globalization culture="en-US" uiCulture="en-US" requestEncoding="utf-8" responseEncoding="utf-8" fileEncoding="utf-8" /></code>	Application root folder
ASPX	Define the appearance and functionality of each page in the Flowtime website. For example, the <code>MyTasks.aspx</code> file defines the Inbox page.	Application root folder
Control Template	The Flowtime website is assembled from pages that host the user controls. For example, <code>UserMessagesGridControl.ascx</code> is used to display the User Messages grid. Configuration example: View article on how to add the form viewer control to a Flowtime page.	<code>[Site physical path]_CONTROLTEMPLATES\Flowtime</code>
Default.config	Defines the navigation menu links in the Flowtime website. You can edit links, set the URL of each link, and add links to new pages.	<code>[Site physical path]\Shared Resources\Components\Flowtime\Config\Portal\Commands</code>
ServiceMetada config	Set of config files that define the display of columns for each grid. For example, the setting for Messages grid columns is defined in the <code>MessagesServiceMetadata.Portal.Config</code> file. Files with the word <i>Portal</i> in the file name are the files that define grids in the Flowtime website.	<code>[Site physical path]\Shared Resources\Components\Flowtime\Config\ServiceMetadata</code>

Add a new page

1. Create an ASPX file in the site root folder.
2. Rename the file using a meaningful name.
3. Open and edit the file.

Example

In this example, we display the Group Messages Grid control and Form Viewer control on the new page MyPage.aspx. The Form Viewer control enables you to add any form view to a page without referencing a specific process. The sections we added are highlighted.

For the controls to work, you need to add a `<Register>` tag for each control on the page that you add. You can find these links in the Flowtime Application in the IIS.

IMPORTANT

If you add custom code to the page, set it to inherit the *PortalBasePage* class at `PNMsoft.Sequence.Runtime.Web.UI.dll` .

```
<%@ Page Title="" Language="C#" CodeBehind="Default.aspx.cs"
Inherits="PNMsoft.Sequence.Runtime.Web.UI.Pages.Default" AutoEventWireup="true"
%>
<%@ Register TagPrefix="sq" TagName="GroupMessagesGridControl"
Src="" %>
<%@ Register TagPrefix="sq" TagName="FormViewerControl"
Src="" %>
<asp:Content ID="Content" ContentPlaceHolderID="content" runat="server">
  <div class="sqpt-container">

    <div class="sqpt-grid-control">
      <sq:GroupMessagesGridControl ID="GroupMessagesGridControl"
      PageSize="5" DisplayVariables="false"
      runat="server" />
    </div>

    <sq:FormViewerControl ID="ReportsView"
    FormVirtualPath="~/SequenceForms/wf/Test/Test/Form1/DefaultView.ascx"
    runat="server" />
  </div>
</asp:Content>
```

Add a page that includes a dashboard

1. Access the ASPX file that you created in the root folder.
2. Add the following AnalyticsDashboardControl declaration.

```
<%@ Register TagPrefix="sq" TagName="AnalyticsDashboardControl"
Src="" %>
```

3. Add the following markup to the page.

```
<asp:Content ID="Content" ContentPlaceHolderID="Content" runat="server">
  <div class="sqpt-container">
    <sq:AnalyticsDashboardControl runat="server" ID="AnalyticsDashboardControl" Transport="flowtime" The
meName="Default" Mode="Auto"></sq:AnalyticsDashboardControl>
  </div>
</asp:Content>
```

- Configure the following properties.

Property	Description
DataStreamName	Name of the dashboard suite you are using.
Layout	Name of the dashboard from the dashboard suite that you want to display
Transport	By default, the Transport is <i>flowtime</i> . Only advanced users should modify this property.

- (Optional) Configure the follow properties as necessary.

Property	Description
ThemeName	By default, ThemeName is <i>Default</i> . The other valid value is <i>Compact</i> .
Mode	By default, Mode is <i>Auto</i> . The other valid values are <i>Fixed</i> and <i>Fluid</i> . This affects the dashboard layout, whether it is fixed, fluid, or set automatically.

- (Optional) Advanced users can configure the following properties as necessary.

Table Title

Property	Description
Parameters	Can include different context parameters, such as <i>SolutionId</i> of a <i>HotOperations</i> solution, and so on. For this to work, users should inherit the code-behind from specific classes.
DataStreamsURL	Points to the data stream's Web service.

The final ASPX file contains the following markup.

```
<asp:Content ID="Content" ContentPlaceHolderID="Content" runat="server">
  <div class="sqpt-container">
    <sq:AnalyticsDashboardControl runat="server" ID="AnalyticsDashboardControl"
    Transport="flowtime" ThemeName="Default" Mode="Auto"></sq:AnalyticsDashboardControl>
  </div>
</asp:Content>
```

Add a link to navigation menu

In this example, you add a link to a custom **My Tasks** page.

- Open the `Default.config` file.
- Under `<ControlTable> <Menu Id="Portal"><Items>`, add a new menu to the `<MenuItem>` section, as follows.

```
<MenuItem
  Id="MyCustomTasksPage"
  Command="MyCustomTasksPage"
  Text="My Page"
  Tooltip="MY Page">
```

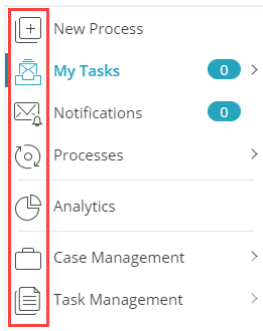
- Under `<CommandTable>`, add a new `NavigateToUrlCommand` section with the URL of the target page.

```
<NavigateToUrlCommand
  Id="MyCustomTasksPage"
  Url="~/MyCustomTasksPage.aspx">
</NavigateToUrlCommand>
```

- Perform IIS Reset, and refresh the homepage.

Set an image for the menu item

Each menu item includes an image. You define the image attribute in the `MenuItem` section.



If you don't define a specific image, a default image is displayed next to the menu item.

You can use one of the following methods to define the menu item image property:

- A regular image path.
For example: `Image="Flowtime/Images/Menu/Processes.png"`
- An icon ID taken from [this list](#).
For example: `Image="sqpi-Airplane"`

Reorder menu items

You can change the order of the menu items, by adding the `Sequence` attribute.

```
<MenuItem
  Id="Processes"
  Text="{ $resources.Processes }"
  Tooltip="{ $resources.Processes }"
  Image="sqpi-Processes"
  SubMenuId="Processes"
  Sequence="1">
</MenuItem>
```

Add a submenu

The Flowtime site menu can include up to two menu levels.

In this example, you add two submenu items to the `Processes` menu item.

- Open the `Default.config` file.
- In a new or existing `MenuItem` section, add the property `SubMenuId="Processes"`.
This setting links the submenu to the correct menu item.

```

<MenuItem
  Id="Processes"
  Text="{ $resources.Processes}"
  Tooltip="{ $resources.Processes}"
  Image="sqpi-Processes"
  SubMenuId="Processes">
</MenuItem>

```

- To define the submenu, add a new Menu tag that includes the new submenu items.
For example:

```

<Menu Id="Processes">
  <Items>
    <MenuItem
      Id="ProcessesIStarted"
      Command="ProcessesIStartedCommand"
      Text="{ $resources.ProcessesIStartedLower}"
      Tooltip="{ $resources.ProcessesIStartedLower}">
    </MenuItem>
    <MenuItem
      Id="ActiveProcesses"
      Command="ActiveProcessesCommand"
      Text="{ $resources.ActiveProcesses}"
      Tooltip="{ $resources.ActiveProcesses}">
    </MenuItem>
  </Items>
</Menu>

```

NOTE

Make sure that the command that you specify exists under the "CommandTable" tag.

- To highlight the selected submenu item, on the `MyTasks.aspx` page, add the following properties:
 - `MenuId`
 - `SelectedItemId`

Example

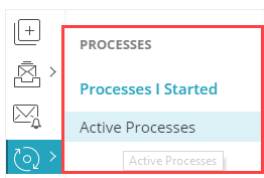
```

<%@ Page Title="My Tasks" ResourceTitle="MyTasks" IconId="sqpi-MyTasks" MenuId="Portal" SubMenuId="MyTasks" SelectedMenuItemId="ProcessesIStarted" SelectedSubMenuId="MyTasks"

```

Resulting menu

Following the configuration described in the example above, the resulting submenu should look like this:



Edit an existing page

In this example, you add a new grid to the existing **MyTasks** page. You will name the grid **MyCustomTasks** page, and then replace the link in the navigation menu to the edited page.

1. Make a copy of the page you want to edit in the root folder.
For example, copy the `MyTasks.aspx` file to the `MyCustomTasksPage.aspx` file.
2. Edit the file as necessary. In the example below, we added a new user control to the `MyTasks.aspx` file. The added sections are highlighted.

```

<%@ Page Title="" Language="C#" MenuId="Portal" SelectedMenuItemId="MyPage" AutoEventWireup="false"
CodeBehind="MyTasks.aspx.cs"
Inherits="PNMsoft.Sequence.Runtime.Web.UI.Pages.MyTasks" %>
<%@ Register TagPrefix="sq" TagName="UserMessagesGridControl"
Src="" %>
<%@ Register TagPrefix="sq" TagName="GroupMessagesGridControl"
Src="" %>
<asp:Content ID="Content" ContentPlaceHolderID="content" runat="server">
<div class="sqpt-container">
<div class="sqpt-grid-control">
<sq:UserMessagesGridControl ID="UserMessagesGridControl"
PageSize="30" DisplayVariables="true" ConfigName="Portal"
runat="server" />
</div>
<div class="sqpt-grid-control">
<sq:GroupMessagesGridControl ID="GroupMessagesGridControl"
PageSize="5" DisplayVariables="false"
runat="server" />
</div>
</asp:Content>

```

3. Replace the link on the navigation menu to direct to the new page that you have created.
4. Open the `Default.config` file, which is located at `C:\Program Files\PNMsoft\Shared Resources\Components\Flowtime\Config\Portal\Commands`, and edit the `<NavigateToUrlCommand>` section as follows:

```

<NavigateToUrlCommand
Id="MyTasks"
Url="~/MyCustomTasksPage.aspx">
</NavigateToUrlCommand>

```

Edit grid layout

The **Layout** property in the grid control of the ASPX file controls the column and table widths.

Values	Description
Auto	Sets column width to fit the widest unbreakable content in the cells. This option can run slower than the Fixed option, because all content in the grid needs to be read before determining the final layout.
Fixed	Default value. The default column width is 130 pixels. You can overwrite the default width in the config file. Horizontal layout depends only on the table's width and the width of the columns, not the contents of the cells. The Fixed layout enables a browser to lay out the table faster than the automatic table layout. The browser can display the table after the first row is read. It is important to be aware of column widths configured with global and workflow variables.

Edit grid scrollbar behavior

You use the **Adaptive** property in the grid control of the ASPX file to determine the grid's scroll bar.

By default, the Adaptive property is set to *True*, and the layout is set to *Fixed*.

Values	Description
True	Default value. Both the header and the table content have their own HTML tables.
False	The grid does not have its own scroll bar (but the page is still scrollable).

Edit grid height

You use the **AdaptiveHeight** property in the grid control of the ASPX file to determine the grid's height. By default, the AdaptiveHeight property is set to -1.

Value	Description
Number >0	Set the height of the grid (numbers of pixels).
0	Grid height matches the pagesize property.
-1	Grid height adapts to the size of the page window. >>>The grid auto-resizes with the page. This option is preferable for pages with a single grid.

Customize column width

You can use the **Width** property in the specific grid's ServiceMetada config to control the width of several types of columns, such as DataBound columns and GlobalVariables columns.

For example, to define the width of the columns for local and global variables in the Process page grid, add the `<workflowvariablescolumns>` tag to the grid's `ServiceMetadata.config` file.

Make sure that you do not add both the `<workflowvariablescolumns>` and `<globalvariablescolumns>` tags.

Example

```
<workflowvariablescolumns>
  <ColumnTemplates>
  <VariableDataBoundViewColumn
    DataField="Variable2Name" Width="200px"/>
  </ColumnTempates>
</workflowvariablescolumns>
```

Configure Tile display

The following grids support Tile Display:

- Flowtime
 - My Tasks
 - Notifications

- Hot Operations
 - Team Member: Task List
 - Team Leader: Task List
 - Operations Manager: Case List
 - Case Search

You can customize the layout of the collapsed and expanded views, by changing the number or columns and their width, and the number of rows per column. You can set up to 12 columns, depending on the width required for each column.

The order in which the data is distributed within the layout is determined by the grid's `ServiceMetadata config` files. Users can also determine the order of the displayed data by reordering the columns in Flowtime.

Grid layout files

For the grids that support Tile display, you configure the layout by editing the following HTML files:

- `grid-tile.html` : configure the layout of the collapsed tile.
- `grid-expanded-tile.html` : configure the layout of the expanded tile.

Component	File location
Flowtime	<code>[rootfolder]\Cora SeQuence\Flowtime\Shared Resources\Components\Flowtime\Templates\ProcessTasks</code>
Hot Operations	<code>[rootfolder]\Cora SeQuence\Flowtime\Shared Resources\Components\HotOperations\Flowtime\Templates\TeamMember</code> <code>[rootfolder]\Cora SeQuence\Flowtime\Shared Resources\Components\HotOperations\Flowtime\Templates\OpsManager</code> <code>[rootfolder]\Cora SeQuence\Flowtime\Shared Resources\Components\HotOperations\Flowtime\Templates\TeamLeader</code> <code>[rootfolder]\Cora SeQuence\Flowtime\Shared Resources\Components\HotOperations\Flowtime\Templates\SearchResults</code>

NOTE

As default files are overwritten with every upgrade, create a custom template based on the HTML files provided.

Tile display application variables

You can set up application variables to avoid losing any changes made to the tile display templates when the system is upgraded. Application variables are stored in the database and define the path of the tile display template.

Component	Key	Value
-----------	-----	-------

Component	Key	Value
Flowtime	Portal.Grid.TileTemplate.ProcessTasks	Path to the file location. For example, the path to the template of the Team Member grid:
	Portal.Grid.ExpandedTileTemplate.ProcessTasks	
Hot Operations	Portal.HotOperations.Grid.TileTemplate.OpsManager	[rootfolder]\Core Components\Flowtime\Shared Resources\Components\HotOperations\Flowti me\Templates\TeamMember\MYgrid- tile.html
	Portal.HotOperations.Grid.ExpandedTileTemplate.OpsMa nager	
	Portal.HotOperations.Grid.TileTemplate.TeamLeader	
	Portal.HotOperations.Grid.ExpandedTileTemplate.TeamL eader	
	Portal.HotOperations.Grid.TileTemplate.TeamMember	
	Portal.HotOperations.Grid.ExpandedTileTemplate.TeamM ember	
	Portal.Grid.TileTemplate.SearchResults	
	Portal.Grid.ExpandedTileTemplate.SearchResults	

Example of collapsed mode for the ProcessTasks grid

In the example, the first five columns of the tile include one row each with different widths, and then four columns with three rows each. The first column includes the "expand" arrow.

Where:

@t (key and value): represents a cell in the grid, that is, the value of the column with its caption.

sqf-col-xs-<integer>: defines the width of the column according to the grid system.

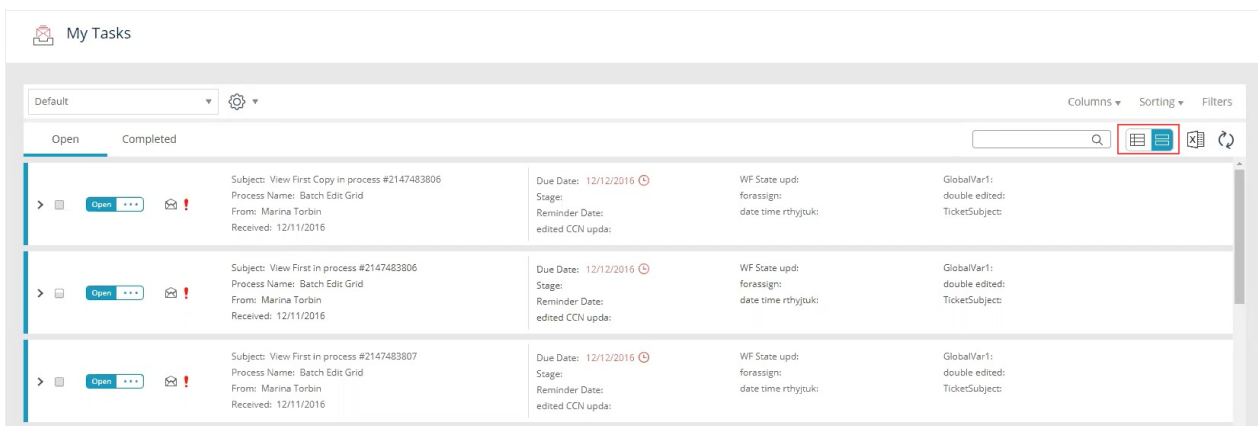
For information on how to customize grid layouts, see [this article](#).

```

<div class="sqf-layout-fluid">
  <div class="sqf-container">
    <div class="sqf-row ft-flex">
      <div class="sqf-col-xs-2 ft-flex ft-flex-v-center" style="flex: 0 0 260px;">
        <div class="sqf-col-xs-1">
          <div class="expand"></div>
        </div>
        <div class="sqf-col-xs-2">@t</div>
        <div class="sqf-col-xs-5">@t</div>
        <div class="sqf-col-xs-1">@t</div>
        <div class="sqf-col-xs-1">@t</div>
        <div class="sqf-col-xs-1">@t</div>
      </div>
      <div class="sqf-col-xs-3 ft-col-b-r">
        @t@t@t@t
      </div>
      <div class="sqf-col-xs-2">
        @t@t@t@t
      </div>
      <div class="sqf-col-xs-2">
        @t@t@t
      </div>
      <div class="sqf-col-xs-2">
        @t@t@t
      </div>
    </div>
  </div>
</div>

```

Above collapsed tile layout looks as follows on Flowtime:



Example of expanded mode for the ProcessTasks grid

In the example, the expanded area of the tile layout include two columns, with the same width each. Where:

i +=<integer>: determines the number of columns.

sqf-col-xs<integer>: defines the width of the column.

```

<div class="sqf-layout-fluid">
  <div class="sqf-container expanded">
    # for (i = count; i < cells.length; i += 2) { #
      <div class="sqf-row">
        <div class="sqf-col-xs-6">@t</div><div class="sqf-col-xs-6">@t</div>
        </div>
      # } #
    </div>
  </div>
</div>

```

Rebranding the Flowtime website

Cora SeQUENCE V8.7 introduced a new Flowtime theme: Cora.

You can customize the Cora theme using two sets of variables: one that changes the look and feel of the Flowtime website and another that changes elements in task forms.

Among other things, you can edit the Flowtime design variables to change:

- Site's logo
- Font colors
- Background color, or add image as background
- Page icons

For more information, see the article [Redesign Flowtime using variables](#).

Hot Operations

Menu item configuration

Hot Operations menu items display based on the user roles.

You set the Hot Operations display rules in the `HotOperations.config` file, under `<MenuItem> <DisplayRules>` .

Setting	Description
<code><ho:IsOpsManagerRule /></code>	Displays the relevant menu item to the Hot Operations Manager role.
<code><ho:IsTeamLeaderRule /></code>	Displays the relevant menu item to the Hot Operations Team Leader role.

Example

```

<MenuItem
  Id="CaseManagement"
  Text="Case Management"
  Tooltip="Case Management"
  Image="sqpi-CaseManagement"
  SubMenuId="HotOperationsManager">
  <DisplayRules>
    <ho:IsOpsManagerRule />
  </DisplayRules>
</MenuItem>

```

Change your solutions' URL

You can set up an application variable to control the menu solutions displayed in the **My Tasks** submenu.

Key	Description	Value
Portal.HotOperations.TeamMember.DefaultPage	Determines the Team Member's default task list page of a solution under My Tasks.	URL
Portal.HotOperations.TeamMember.SolutionsMenuId	Determines the menu to which the solutions' task list will be added.	Menu Id

Hide the Process Tasks menu item

To hide the **Process Tasks** menu item from the My Tasks submenu, do the following:

1. In the `HotOperations.config` file, remove the **MyTasks** menu item from **TaskList** menu.

Hot Operations notifications

Manual actions performed by the Team Leader and Hot Operations Manager are now displayed on a new notifications pop-up window. You can control whether notifications appear or not, and for how long they are displayed.

Attribute	Description	Value
DisplayItemsProgressBox	Determines whether or not the notification pop-up window displays when the user performs an action on the grid items. For example, Reassign or Reallocate.	True (default) False
ItemsProgressBoxVisibilityDuration	Determines the duration, in seconds, the notification pop-up window is visible before it fades.	Numeric 10 (default)

To set up the notifications pop-up window, edit the following files with the required attributes.

File	Section	Attribute
[Site physical path]\HotOperations\OpsManager\CasesList.aspx	<sq:TeamLeaderTasksGridControl>	DisplayItemsProgressBox
[Site physical path]\HotOperations\OpsManager\AllocationBoard.aspx	<sq:TeamLeaderTasksGridControl>	ItemsProgressBoxVisibilityDuration
[Site physical path]\HotOperations\TeamLeader\TasksList.aspx	<sq:TeamLeaderTasksGridControl>	
[Site physical path]\HotOperations\TeamLeader\AssignmentBoard.aspx	<sq:TeamLeaderTasksGridControl>	

Example

```
<div class="sqpt-container">
  <sq:TeamLeaderTasksGridControl ID="TeamLeaderTasksGridControl" runat="server" DisplayItemsProgressBox
  ="True" ItemsProgressBoxVisibilityDuration="5" AssignedDynamicTasksViewType="All" DisplayVariables="true" Page
  Size="20" />
</div>
```

Translate custom Flowtime components

- Add following attribute to the Flowtime web.config file:

```
<resources resXFilesLocation="Resources" externalGlobalResourceName="CustomResourceFileName" />
```

- Under *Shared Resources/Resources/Sequence* add translation files.
(CustomResourceFileName.resx or CustomResourceFileName.es.resx)
- Use the following expression to get resource value:

```
{ $resources.KeyName }
```

Example

```
<MenuItem Id="CoraOpsProcessesIStarted"
  Command="CoraOpsProcessesIStartedCommand"
  Text="{ $resources.CasesIStarted }"
  Tooltip="{ $resources.CasesIStarted }"
  Image="sqpi-Processes">
</MenuItem>
```

V9.1

You configure the Flowtime site to provide end users with the smooth functionality and usability. You can customize the pages to comply with your organization's branding policies, add or edit existing pages, set up functionality that fit existing processes, and configure new menus and submenus.

Configuration files

After you install the Flowtime site, you can configure the following files:

File type	Description	Location
web.config	Controls settings, such as culture, theme, and database connection. For example, you can set the Flowtime website language by editing the following key: <pre><globalization culture="en-US" uiCulture="en-US" requestEncoding="utf-8" responseEncoding="utf-8" fileEncoding="utf-8" /></pre>	Application root folder
ASPX	Define the appearance and functionality of each page in the Flowtime website. For example, the <code>MyTasks.aspx</code> file defines the Inbox page.	Application root folder

File type	Description	Location
Control Template	The Flowtime website is assembled from pages that host the user controls. For example, <code>UserMessagesGridControl.ascx</code> is used to display the User Messages grid. Configuration example: View article on how to add the form viewer control to a Flowtime page.	[Site physical path]_CONTROLTEMPLATES\Flowtime
Default.config	Defines the navigation menu links in the Flowtime website. You can edit links, set the URL of each link, and add links to new pages.	[Site physical path]\Shared Resources\Components\Flowtime\Config\Portal\Commands
ServiceMetada config	Set of config files that define the display of columns for each grid. For example, the setting for Messages grid columns is defined in the <code>MessagesServiceMetadata.Portal.Config</code> file. Files with the word <i>Portal</i> in the file name are the files that define grids in the Flowtime website.	[Site physical path]\Shared Resources\Components\Flowtime\Config\ServiceMetadata

Add a new page

1. Create an ASPX file in the site root folder.
2. Rename the file using a meaningful name.
3. Open and edit the file.

Example

In this example, we display the Group Messages Grid control and Form Viewer control on the new page `MyPage.aspx`. The Form Viewer control enables you to add any form view to a page without referencing a specific process. The sections we added are highlighted.

For the controls to work, you need to add a `<Register>` tag for each control on the page that you add. You can find these links in the Flowtime Application in the IIS.

IMPORTANT

If you add custom code to the page, set it to inherit the `PortalBasePage` class at `PNMsoft.Sequence.Runtime.Web.UI.dll`.

```

<%@ Page Title="" Language="C#" CodeBehind="Default.aspx.cs"
Inherits="PNMsoft.Sequence.Runtime.Web.UI.Pages.Default" AutoEventWireup="true"
%>
<%@ Register TagPrefix="sq" TagName="GroupMessagesGridControl"
Src="" %>
<%@ Register TagPrefix="sq" TagName="FormViewerControl"
Src="" %>
<asp:Content ID="Content" ContentPlaceHolderID="content" runat="server">
  <div class="sqpt-container">

    <div class="sqpt-grid-control">
      <sq:GroupMessagesGridControl ID="GroupMessagesGridControl"
      PageSize="5" DisplayVariables="false"
      runat="server" />
    </div>

    <sq:FormViewerControl ID="ReportsView"
    FormVirtualPath="~/SequenceForms/wf/Test/Test/Form1/DefaultView.ascx"
    runat="server" />
  </div>
</asp:Content>

```

Add a page that includes a dashboard

1. Access the ASPX file that you created in the root folder.
2. Add the following AnalyticsDashboardControl declaration.

```

<%@ Register TagPrefix="sq" TagName="AnalyticsDashboardControl"
Src="" %>

```

3. Add the following markup to the page.

```

<asp:Content ID="Content" ContentPlaceHolderID="Content" runat="server">
  <div class="sqpt-container">
    <sq:AnalyticsDashboardControl runat="server" ID="AnalyticsDashboardControl" Transport="flowtime" The
meName="Default" Mode="Auto"></sq:AnalyticsDashboardControl>
  </div>
</asp:Content>

```

4. Configure the following properties.

Property	Description
DataStreamName	Name of the dashboard suite you are using.
Layout	Name of the dashboard from the dashboard suite that you want to display
Transport	By default, the Transport is <i>flowtime</i> . Only advanced users should modify this property.

5. (Optional) Configure the follow properties as necessary.

Property	Description
ThemeName	By default, ThemeName is <i>Default</i> . The other valid value is <i>Compact</i> .

Property	Description
Mode	By default, Mode is Auto. The other valid values are <i>Fixed</i> and <i>Fluid</i> . This affects the dashboard layout, whether it is fixed, fluid, or set automatically.

6. (Optional) Advanced users can configure the following properties as necessary.

Table Title

Property	Description
Parameters	Can include different context parameters, such as <i>SolutionId</i> of a <i>HotOperations</i> solution, and so on. For this to work, users should inherit the code-behind from specific classes.
DataStreamURL	Points to the data stream's Web service.

The final ASPX file contains the following markup.

```
<asp:Content ID="Content" ContentPlaceHolderID="Content" runat="server">
  <div class="sqpt-container">
    <sq:AnalyticsDashboardControl runat="server" ID="AnalyticsDashboardControl"
    Transport="flowtime" ThemeName="Default" Mode="Auto"></sq:AnalyticsDashboardControl>
  </div>
</asp:Content>
```

Add a link to navigation menu

In this example, you add a link to a custom **My Tasks** page.

1. Open the `Default.config` file.
2. Under `<ControlTable> <Menu Id="Portal"><Items>`, add a new menu to the `<MenuItem>` section, as follows.

```
<MenuItem
  Id="MyCustomTasksPage"
  Command="MyCustomTasksPage"
  Text="My Page"
  Tooltip="MY Page">
```

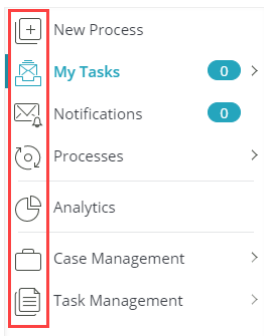
3. Under `<CommandTable>`, add a new `NavigateToUrlCommand` section with the URL of the target page.

```
<NavigateToUrlCommand
  Id="MyCustomTasksPage"
  Url="~/MyCustomTasksPage.aspx">
</NavigateToUrlCommand>
```

4. Perform IIS Reset, and refresh the homepage.

Set an image for the menu item

Each menu item includes an image. You define the image attribute in the `MenuItem` section.



If you don't define a specific image, a default image is displayed next to the menu item.

You can use one of the following methods to define the menu item image property:

- A regular image path.
For example: `Image="Flowtime/Images/Menu/Processes.png"`
- An icon ID taken from [this list](#).
For example: `Image="sqpi-Airplane"`

Reorder menu items

You can change the order of the menu items, by adding the `Sequence` attribute.

```
<MenuItem
  Id="Processes"
  Text="{ $resources.Processes}"
  Tooltip="{ $resources.Processes}"
  Image="sqpi-Processes"
  SubMenuId="Processes"
  Sequence="1">
</MenuItem>
```

Add a submenu

The Flowtime site menu can include up to two menu levels.

In this example, you add two submenu items to the `Processes` menu item.

1. Open the `Default.config` file.
2. In a new or existing `MenuItem` section, add the property `SubMenuId="Processes"`.
This setting links the submenu to the correct menu item.

```
<MenuItem
  Id="Processes"
  Text="{ $resources.Processes}"
  Tooltip="{ $resources.Processes}"
  Image="sqpi-Processes"
  SubMenuId="Processes">
</MenuItem>
```

2. To define the submenu, add a new `Menu` tag that includes the new submenu items.
For example:

```

<Menu Id="Processes">
  <Items>
    <MenuItem
      Id="ProcessesIStarted"
      Command="ProcessesIStartedCommand"
      Text="{ $resources.ProcessesIStartedLower}"
      Tooltip="{ $resources.ProcessesIStartedLower}">
    </MenuItem>
    <MenuItem
      Id="ActiveProcesses"
      Command="ActiveProcessesCommand"
      Text="{ $resources.ActiveProcesses}"
      Tooltip="{ $resources.ActiveProcesses}">
    </MenuItem>
  </Items>
</Menu>

```

NOTE

Make sure that the command that you specify exists under the `CommandTable` tag.

3. To highlight the selected submenu item, on the `MyTasks.aspx` page, add the following properties:

- `MenuId`
- `SelectedItemId`

Example

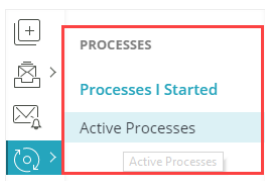
```

<%@ Page Title="My Tasks" ResourceTitle="MyTasks" IconId="sqpi-MyTasks" MenuId="Portal" SubMenuId="MyTasks" SelectedMenuItemId="ProcessesIStarted" SelectedSubMenuItemId="MyTasks"

```

Resulting menu

Following the configuration described in the example above, the resulting submenu should look like this:



Edit an existing page

In this example, you add a new grid to the existing **MyTasks** page. You will name the grid **MyCustomTasks** page, and then replace the link in the navigation menu to the edited page.

1. Make a copy of the page you want to edit in the root folder.
For example, copy the `MyTasks.aspx` file to the `MyCustomTasksPage.aspx` file.
2. Edit the file as necessary. In the example below, we added a new user control to the `MyTasks.aspx` file. The added sections are highlighted.

```

<%@ Page Title="" Language="C#" MenuId="Portal" SelectedMenuItemId="MyPage" AutoEventWireup="false"
"
CodeBehind="MyTasks.aspx.cs"
Inherits="PNMsoft.Sequence.Runtime.Web.UI.Pages.MyTasks" %>
<%@ Register TagPrefix="sq" TagName="UserMessagesGridControl"
Src="" %>
<%@ Register TagPrefix="sq" TagName="GroupMessagesGridControl"
Src="" %>
<asp:Content ID="Content" ContentPlaceHolderID="content" runat="server">
<div class="sqpt-container">
<div class="sqpt-grid-control">
<sq:UserMessagesGridControl ID="UserMessagesGridControl"
PageSize="30" DisplayVariables="true" ConfigName="Portal"
runat="server" />
</div>
<div class="sqpt-grid-control">
<sq:GroupMessagesGridControl ID="GroupMessagesGridControl"
PageSize="5" DisplayVariables="false"
runat="server" />
</div>
</asp:Content>

```

3. Replace the link on the navigation menu to direct to the new page that you have created.
4. Open the `Default.config` file, which is located at `C:\Program Files\PNMsoft\Shared Resources\Components\Flowtime\Config\Portal\Commands`, and edit the `<NavigateToUrlCommand>` section as follows:

```

<NavigateToUrlCommand
Id="MyTasks"
Url="~/MyCustomTasksPage.aspx">
</NavigateToUrlCommand>

```

Edit grid layout

The **Layout** property in the grid control of the ASPX file controls the column and table widths.

Values	Description
Auto	Sets column width to fit the widest unbreakable content in the cells. This option can run slower than the Fixed option, because all content in the grid needs to be read before determining the final layout.
Fixed	Default value. The default column width is 130 pixels. You can overwrite the default width in the config file. Horizontal layout depends only on the table's width and the width of the columns, not the contents of the cells. The Fixed layout enables a browser to lay out the table faster than the automatic table layout. The browser can display the table after the first row is read. It is important to be aware of column widths configured with global and workflow variables.

Edit grid scrollbar behavior

You use the **Adaptive** property in the grid control of the ASPX file to determine the grid's scroll bar.

By default, the Adaptive property is set to *True*, and the layout is set to *Fixed*.

Values	Description
True	Default value. Both the header and the table content have their own HTML tables.
False	The grid does not have its own scroll bar (but the page is still scrollable).

Edit grid height

You use the **AdaptiveHeight** property in the grid control of the ASPX file to determine the grid's height. By default, the AdaptiveHeight property is set to -1.

Value	Description
Number >0	Set the height of the grid (numbers of pixels).
0	Grid height matches the pagesize property.
-1	Grid height adapts to the size of the page window. >>>The grid auto-resizes with the page. This option is preferable for pages with a single grid.

Customize column width

You can use the **Width** property in the specific grid's ServiceMetada config to control the width of several types of columns, such as DataBound columns and GlobalVariables columns.

For example, to define the width of the columns for local and global variables in the Process page grid, add the `<workflowvariablescolumns>` tag to the grid's `ServiceMetadata.config` file.

Make sure that you do not add both the `<workflowvariablescolumns>` and `<globalvariablescolumns>` tags.

Example

```
<workflowvariablescolumns>
  <ColumnTemplates>
    <VariableDataBoundViewColumn
      DataField="Variable2Name" Width="200px"/>
    </ColumnTempates>
  </workflowvariablescolumns>
```

Rebranding the Flowtime website

Cora SeQuence V8.7 introduced a new Flowtime theme: Cora.

You can customize the Cora theme using two sets of variables: one that changes the look and feel of the Flowtime website and another that changes elements in task forms.

Among other things, you can edit the Flowtime design variables to change:

- Site's logo
- Font colors
- Background color, or add image as background

- Page icons

For more information, see the article [Redesign Flowtime using variables](#).

Hot Operations

Menu item configuration

Hot Operations menu items display based on the user roles.

You set the Hot Operations display rules in the `HotOperations.config` file, under `<MenuItem> <DisplayRules>`.

Setting	Description
<code><ho:IsOpsManagerRule /></code>	Displays the relevant menu item to the Hot Operations Manager role.
<code><ho:IsTeamLeaderRule /></code>	Displays the relevant menu item to the Hot Operations Team Leader role.

Example

```
<MenuItem
  Id="CaseManagement"
  Text="Case Management"
  Tooltip="Case Management"
  Image="sqpi-CaseManagement"
  SubMenuId="HotOperationsManager">
  <DisplayRules>
    <ho:IsOpsManagerRule />
  </DisplayRules>
</MenuItem>
```

Change your solutions' URL

You can set up an application variable to control the menu solutions displayed in the **My Tasks** submenu.

Key	Description	Value
Portal.HotOperations.TeamMember.DefaultPage	Determines the Team Member's default task list page of a solution under My Tasks.	URL
Portal.HotOperations.TeamMember.SolutionsMenuId	Determines the menu to which the solutions' task list will be added.	Menu Id

Hide the Process Tasks menu item

To hide the **Process Tasks** menu item from the My Tasks submenu, do the following:

1. In the `HotOperations.config` file, remove the **MyTasks** menu item from **TaskList** menu.

Hot Operations notifications

Manual actions performed by the Team Leader and Hot Operations Manager are now displayed on a new notifications pop-up window. You can control whether notifications appear or not, and for how long they are displayed.

Attribute	Description	Value
DisplayItemsProgressBox	Determines whether or not the notification pop-up window displays when the user performs an action on the grid items. For example, Reassign or Reallocate.	True (default) False
ItemsProgressBoxVisibilityDuration	Determines the duration, in seconds, the notification pop-up window is visible before it fades.	Numeric 10 (default)

To set up the notifications pop-up window, edit the following files with the required attributes.

File	Section	Attribute
[Site physical path]\HotOperations\OpsManager\CasesList.aspx	<sq:TeamLeaderTasksGridControl>	DisplayItemsProgressBox
[Site physical path]\HotOperations\OpsManager\AllocationBoard.aspx	<sq:TeamLeaderTasksGridControl>	ItemsProgressBoxVisibilityDuration
[Site physical path]\HotOperations\TeamLeader\TasksList.aspx	<sq:TeamLeaderTasksGridControl>	
[Site physical path]\HotOperations\TeamLeader\AssignmentBoard.aspx	<sq:TeamLeaderTasksGridControl>	

Example

```
<div class="sqpt-container">
  <sq:TeamLeaderTasksGridControl ID="TeamLeaderTasksGridControl" runat="server" DisplayItemsProgressBox="True" ItemsProgressBoxVisibilityDuration="5" AssignedDynamicTasksViewType="All" DisplayVariables="true" PageSize="20" />
</div>
```

V8.7

You configure the Flowtime website to provide end users with the smooth functionality and usability. You can customize the pages to comply with your organization's branding policies, add or edit existing pages, and set up functionality to fit existing processes.

Configuration files

After you install the Flowtime site, you can configure the following files:

File type	Description	Location
web.config	Controls settings, such as culture, theme, and database connection. For example, you can set the Flowtime website language by editing the following key: <code><globalization culture="en-US" uiCulture="en-US" requestEncoding="utf-8" responseEncoding="utf-8" fileEncoding="utf-8" /></code>	Application root folder
ASPX	Define the appearance and functionality of each page in the Flowtime website. For example, the <code>MyTasks.aspx</code> file defines the Inbox page.	Application root folder
Control Template	The Flowtime website is assembled from pages that host the user controls. For example, <code>UserMessagesGridControl.ascx</code> is used to display the User Messages grid. Configuration example: View article on how to add the form viewer control to a Flowtime page.	<code>{Site Physical Path}\CONTROLTEMPLATES\Flowtime</code>
Default.config	Defines the navigation menu links in the Flowtime website. You can edit links, set the URL of each link, and add links to new pages.	<code>C:\Program Files\PNMsoft\Shared Resources\Components\Flowtime\Config\Portal\Commands</code>
ServiceMetadata.config	Set of config files that define the display of columns for each grid. For example, the setting for Messages grid columns is defined in the <code>MessagesServiceMetadata.Portal.Config</code> file. Files with the word <i>Portal</i> in the file name are the files that define grids in the Flowtime website.	<code>C:\Program Files\PNMsoft\Shared Resources\Components\Flowtime\Config\ServiceMetadata</code>

Add a new page

1. Create an ASPX file in the site root folder.
2. Rename the file using a meaningful name.
3. Open and edit the file.

Example

In this example, we display the Group Messages Grid control and Form Viewer control on the new page `MyPage.aspx`. The Form Viewer control enables you to add any form view to a page without referencing a specific process. The sections we added are highlighted.

For the controls to work, you need to add a `<Register>` tag for each control on the page that you add.

You can find these links in the Flowtime Application in the IIS.

IMPORTANT

If you add custom code to the page, set it to inherit the *PortalBasePage* class at

`PNMsoft.Sequence.Runtime.Web.UI.dll`.

```
<%@ Page Title="" Language="C#" CodeBehind="Default.aspx.cs"
Inherits="PNMsoft.Sequence.Runtime.Web.UI.Pages.Default" AutoEventWireup="true"
%>
<%@ Register TagPrefix="sq" TagName="GroupMessagesGridControl"
Src="" %>
<%@ Register TagPrefix="sq" TagName="FormViewerControl"
Src="" %>
<asp:Content ID="Content" ContentPlaceHolderID="content" runat="server">
  <div class="sqpt-container">

    <div class="sqpt-grid-control">
      <sq:GroupMessagesGridControl ID="GroupMessagesGridControl"
      PageSize="5" DisplayVariables="false"
      runat="server" />
    </div>

    <sq:FormViewerControl ID="ReportsView"
    FormVirtualPath="~/SequenceForms/wf/Test/Test/Form1/DefaultView.ascx"
    runat="server" />
  </div>
</asp:Content>
```

Add a page that includes a dashboard

1. Access the ASPX file that you created in the root folder.
2. Add the following `AnalyticsDashboardControl` declaration.

```
<%@ Register TagPrefix="sq" TagName="AnalyticsDashboardControl"
Src="" %>
```

3. Add the following markup to the page.

```
<asp:Content ID="Content" ContentPlaceHolderID="Content" runat="server">
  <div class="sqpt-container">
    <sq:AnalyticsDashboardControl runat="server" ID="AnalyticsDashboardControl" Transport="flowtime" The
    meName="Default" Mode="Auto"></sq:AnalyticsDashboardControl>
  </div>
</asp:Content>
```

4. Configure the following properties.

Property	Description
DataStreamName	Name of the dashboard suite you are using.
Layout	Name of the dashboard from the dashboard suite that you want to display
Transport	By default, the Transport is <i>flowtime</i> . Only advanced users should modify this property.

5. (Optional) Configure the follow properties as necessary.

Property	Description
ThemeName	By default, ThemeName is <i>Default</i> . The other valid value is <i>Compact</i> .
Mode	By default, Mode is <i>Auto</i> . The other valid values are <i>Fixed</i> and <i>Fluid</i> . This affects the dashboard layout, whether it is fixed, fluid, or set automatically.

6. (Optional) Advanced users can configure the following properties as necessary.

Table Title

Property	Description
Parameters	Can include different context parameters, such as <i>SolutionId</i> of a <i>HotOperations</i> solution, and so on. For this to work, users should inherit the code-behind from specific classes.
DataStreamURL	Points to the data stream's Web service.

The final ASPX file contains the following markup.

```
<asp:Content ID="Content" ContentPlaceHolderID="Content" runat="server">
  <div class="sqpt-container">
    <sq:AnalyticsDashboardControl runat="server" ID="AnalyticsDashboardControl"
    Transport="flowtime" ThemeName="Default" Mode="Auto"></sq:AnalyticsDashboardControl>
  </div>
</asp:Content>
```

Add a link to navigation menu

In this example, you add a link to a custom **My Tasks** page.

1. Open the `Default.config` file.
2. Under `<ControlTable> <Menu Id="Portal"><Items>`, add a new menu to the `<MenuItem>` section, as follows.

```
<MenuItem
  Id="MyCustomTasksPage"
  Command="MyCustomTasksPage"
  Text="My Page"
  Tooltip="MY Page">
</MenuItem>
```

3. Under `<CommandTable>`, add a new `NavigateToUrlCommand` section with the URL of the target page.

```
<NavigateToUrlCommand
  Id="MyCustomTasksPage"
  Url="~/MyCustomTasksPage.aspx">
</NavigateToUrlCommand>
```

4. Perform IIS Reset, and refresh the homepage.

Edit an existing page

In this example, you edit the existing **MyTasks** page to add a new grid, name it **MyCustomTasks** page,

and then replace the link in the navigation menu to the edited page.

1. Make a copy of the page you want to edit in the root folder. For example, copy the `MyTasks.aspx` file to the `MyCustomTasksPage.aspx` file.
2. Edit the file as necessary. In the example below, we added a new user control to the `MyTasks.aspx` file. The added sections are highlighted.

```
<%@ Page Title="" Language="C#" MenuId="Portal" SelectedMenuItemId="MyPage" AutoEventWireup="false"
CodeBehind="MyTasks.aspx.cs"
Inherits="PNMsoft.Sequence.Runtime.Web.UI.Pages.MyTasks" %>
<%@ Register TagPrefix="sq" TagName="UserMessagesGridControl"
Src="" %>
<%@ Register TagPrefix="sq" TagName="GroupMessagesGridControl"
Src="" %>
<asp:Content ID="Content" ContentPlaceHolderID="content" runat="server">
<div class="sqpt-container">
<div class="sqpt-grid-control">
<sq:UserMessagesGridControl ID="UserMessagesGridControl"
PageSize="30" DisplayVariables="true" ConfigName="Portal"
runat="server" />
</div>
<div class="sqpt-grid-control">
<sq:GroupMessagesGridControl ID="GroupMessagesGridControl"
PageSize="5" DisplayVariables="false"
runat="server" />
</div>
</asp:Content>
```

3. Replace the link on the navigation menu to direct to the new page that you have created.
4. Open the `Default.config` file, which is located at `C:\Program Files\PNMsoft\Shared Resources\Components\Flowtime\Config\Portal\Commands`, and edit the `<NavigateToUrlCommand>` section as follows:

```
<NavigateToUrlCommand
Id="MyTasks"
Url="~/MyCustomTasksPage.aspx">
</NavigateToUrlCommand>
```

Edit grid layout

The **Layout** property in the grid control of the ASPX file controls the column and table widths.

Values	Description
Auto	Sets column width to fit the widest unbreakable content in the cells. This option can run slower than the Fixed option, because all content in the grid needs to be read before determining the final layout.
Fixed	Default value. The default column width is 130 pixels. You can overwrite the default width in the config file. Horizontal layout depends only on the table's width and the width of the columns, not the contents of the cells. The Fixed layout enables a browser to lay out the table faster than the automatic table layout. The browser can display the table after the first row is read. It is important to be aware of column widths configured with global and workflow variables.

Edit grid scrollbar behavior

You use the **Adaptive** property in the grid control of the ASPX file to determine the grid's scroll bar.

By default, the Adaptive property is set to *True*, and the layout is set to *Fixed*.

Values	Description
True	Default value. Both the header and the table content have their own HTML tables.
False	The grid does not have its own scroll bar (but the page is still scrollable).

Edit grid height

You use the **AdaptiveHeight** property in the grid control of the ASPX file to determine the grid's height.

By default, the AdaptiveHeight property is set to -1.

Value	Description
Number >0	Set the height of the grid (numbers of pixels).
0	Grid height matches the pagesize property.
-1	Grid height adapts to the size of the page window. >>>The grid auto-resizes with the page. This option is preferable for pages with a single grid.

Customize column width

You can use the **Width** property in the specific grid's ServiceMetadata config to control the width of several types of columns, such as DataBound columns and GlobalVariables columns.

For example, to define the width of the columns for local and global variables in the Process page grid, add the `<workflowvariablescolumns>` tag to the grid's `ServiceMetadata.config` file.

Make sure that you do not add both the `<workflowvariablescolumns>` and `<globalvariablescolumns>` tags.

Example

```
<workflowvariablescolumns>
  <ColumnTemplates>
  <VariableDataBoundViewColumn
    DataField="Variable2Name" Width="200px"/>
  </ColumnTempates>
</workflowvariablescolumns>
```

Rebranding the Flowtime website

Cora SeQence V8.7 introduced a new Flowtime theme: Cora.

You can customize the Cora theme using two sets of variables: one that changes the look and feel of the Flowtime website and another that changes elements in task forms.

Among other things, you can edit the Flowtime design variables to change:

- Site's logo
- Font colors
- Background color, or add image as background
- Page icons

For more information, see the article [Redesign Flowtime using variables](#).

V8.6

The Flowtime website enables you to provide Flowtime experience and functionality to end users without having to install SharePoint. It provides the option to add and edit pages quite easily, and because it's a website, there is no need to compile.

Understanding File Structure

After you install Flowtime website, there are several files you can modify that enable you to configure and edit the Flowtime website.

Flowtime website files

- web.config
- ASPX
- Control Template
- Default.config
- ServiceMetadata config

web.config files

Allows you to control Flowtime website settings, such as culture, theme, database connection, and so on.

For example, you can set the Flowtime website language by editing the following key:

```
<globalization culture="en-US" uiCulture="en-US" requestEncoding="utf-8" responseEncoding="utf-8" fileEncoding="utf-8" />
```

ASPX files

These files define the appearance and functionality of each page on the Flowtime website. For example, the `Inbox.aspx` file defines the Inbox page.

Control Template files

The Flowtime website is assembled from pages that host the user controls. For example, `UserMessagesGridControl.ascx` is used to display the User Messages grid.

File Location: `{Site Physical Path}\CONTROLTEMPLATES\Flowtime`

Default.config file

This file defines the top menu links in the Flowtime website. Here you can edit links, set the URL of each link, and add links to new pages.

File Location: C:\Program Files\PNMsoft\Shared Resources\Components\Flowtime\Config\Portal\Commands

ServiceMetadata config files

This set of config files define the way columns display for each grid. For example, the setting for Messages grid columns is defined in the `MessagesServiceMetadata.Portal.Config` file. Files with the word *Portal* in the file name are the files that define grids in the Flowtime website.

File Location: C:\Program Files\PNMsoft\Shared Resources\Components\Flowtime\Config\ServiceMetadata

Edit the Flowtime Website

You can easily add pages, edit pages, and make other appearance and functionality changes to the Flowtime website. We'll cover some of the the most common Flowtime website modifications.

Add a new page

1. Create an ASPX file in the site root folder.
2. Rename the file using a meaningful name.
3. Open and edit the file.

Example

In this example we display the Group Messages Grid control and Form Viewer control on the new page `MyPage.aspx`. The Form Viewer control enables you to add any form view to a page without referencing a specific process. The sections we added are highlighted.

For the controls to work, you need to add a `<Register>` tag for each control on the page that you add. You can find these links in the Flowtime Application in the IIS.

IMPORTANT: If you need to add your own code behind for the page, you must inherit from the `PortalBasePage` class from `PNMsoft.Sequence.Runtime.Web.UI.dll`.

```
<%@ Page Title="" Language="C#" CodeBehind="Default.aspx.cs"
Inherits="PNMsoft.Sequence.Runtime.Web.UI.Pages.Default" AutoEventWireup="true"
%>
<%@ Register TagPrefix="sq" TagName="GroupMessagesGridControl"
Src="" %>
<%@ Register TagPrefix="sq" TagName="FormViewerControl"
Src="" %>
<asp:Content ID="Content" ContentPlaceHolderID="content" runat="server">
  <div class="sqpt-container">

    <div class="sqpt-grid-control">
      <sq:GroupMessagesGridControl ID="GroupMessagesGridControl"
      PageSize="5" DisplayVariables="false"
      runat="server" />
    </div>

    <sq:FormViewerControl ID="ReportsView"
    FormVirtualPath="~/SequenceForms/wf/Test/Test/Form1/DefaultView.ascx"
    runat="server" />
  </div>
</asp:Content>
```

Add a Page that Includes a Dashboard

1. Access the ASPX file that you created in the root folder.
2. Add the following AnalyticsDashboardControl declaration.

```
<%@ Register TagPrefix="sq" TagName="AnalyticsDashboardControl"
Src="" %>
```

3. Add the following markup to the page.

```
<asp:Content ID="Content" ContentPlaceHolderID="Content" runat="server">
  <div class="sqpt-container">
    <sq:AnalyticsDashboardControl runat="server" ID="AnalyticsDashboardControl" Transport="flowtime" ThemeName="Default" Mode="Auto"></sq:AnalyticsDashboardControl>
  </div>
</asp:Content>
```

4. Configure the following properties.

Property	Description
DataStreamName	name of the dashboard suite you are using.
Layout	name of the dashboard from the dashboard suite that you want to display
Transport	by default, the Transport is <i>flowtime</i> . Only advanced users should modify this property.

5. (Optional) Configure the follow properties as necessary.

Property	Description
ThemeName	by default, ThemeName is <i>Default</i> . The other valid value is <i>Compact</i> .
Mode	by default, Mode is <i>Auto</i> . The other valid values are <i>Fixed</i> and <i>Fluid</i> . This affects the dashboard layout, whether it is fixed, fluid, or set automatically.

6. (Optional) Advanced users can configure the following properties as necessary.

Table Title

Property	Description
Parameters	can include different context parameters, such as <i>SolutionId</i> of a HotOperations solution, and so on. For this to work, users should inherit the code-behind from specific classes.
DataStreamsURL	points to the data stream's Web service.

The final ASPX file will contain the following markup.

```
<asp:Content ID="Content" ContentPlaceHolderID="Content" runat="server">
  <div class="sqpt-container">
    <sq:AnalyticsDashboardControl runat="server" ID="AnalyticsDashboardControl"
Transport="flowtime" ThemeName="Default" Mode="Auto"></sq:AnalyticsDashboardControl>
  </div>
</asp:Content>
```

Add a Link to the Banner

1. Open the `Default.config` file.
2. Copy a `<MenuItem>` section, and paste it under the existing `<MenuItem>` section.

```
<MenuItem
  Id="Inbox"
  Command="Inbox"
  Text="{ $resources.Inbox}"
  Tooltip="{ $resources.Inbox}">
</MenuItem>
```

3. Edit the values in the `<MenuItem>` section.

```
<MenuItem
  Id="MyPage"
  Command="MyPage"
  Text="My Page"
  Tooltip="MY Page">
</MenuItem>
```

4. Copy a `NavigateToUrlCommand` section and paste it below the existing `NavigateToUrlCommand` section. This section determines the URL of the link.

```
<NavigateToUrlCommand
  Id="Inbox"
  Url="~/Inbox.aspx?MessageInstanceViewType=2">
</NavigateToUrlCommand>
```

5. Edit the values in the `NavigateToUrlCommand` section with the following information.

```
<NavigateToUrlCommand
  Id="MyPage"
  Url="~/MyPage.aspx">
</NavigateToUrlCommand>
```

6. Perform IIS Reset and refresh the homepage.

Edit an Existing Page

1. Make a copy of the page you want to edit in the root folder. For example, copy the `Inbox.aspx` file to the `MyPage.aspx` file.
2. Edit the file as necessary. In the example below, we added a new user control to the `Inbox.aspx` file. The added sections are highlighted.

```

<%@ Page Title="" Language="C#" MenuId="Portal" SelectedMenuItemId="MyPage" AutoEventWireup="false
"
CodeBehind="Inbox.aspx.cs"
Inherits="PNMsoft.Sequence.Runtime.Web.UI.Pages.Inbox" %>
<%@ Register TagPrefix="sq" TagName="InboxSummaryControl"
Src="" %>
<%@ Register TagPrefix="sq" TagName="NewProcessControl"
Src="" %>
<%@ Register TagPrefix="sq" TagName="ProcessesIOwnControl"
Src="" %>
<%@ Register TagPrefix="sq" TagName="UserMessagesGridControl"
Src="" %>
<%@ Register TagPrefix="sq" TagName="GroupMessagesGridControl"
Src="" %>
<asp:Content ID="Content" ContentPlaceHolderID="content" runat="server">
<div class="sqpt-container">
<div class="sqpt-summary-controls">
<div>
<sq:NewProcessControl ID="NewProcessControl"
runat="server" />
</div>
<div>
<sq:InboxSummaryControl ID="InboxSummaryControl" InboxPage="Inbox.aspx"
runat="server" />
</div>
<div>
<sq:ProcessesIOwnControl ID="ProcessesIOwnControl" ProcessesIOwnPage="OpenProcesses.aspx" runat
="server" />
</div>
</div>
<div class="sqpt-grid-control">
<sq:UserMessagesGridControl ID="UserMessagesGridControl"
PageSize="30" DisplayVariables="true" ConfigName="Portal"
runat="server" />
</div>
<div class="sqpt-grid-control">
<sq:GroupMessagesGridControl ID="GroupMessagesGridControl"
PageSize="5" DisplayVariables="false"
runat="server" />
</div>
</div>
</div>

```

3. Change the page reference in the `Default.config` file, which is located at `C:\Program Files\PNMsoft\Shared Resources\Components\Flowtime\Config\Portal\Commands`.

```

<NavigateToUrlCommand
Id="Inbox"
Url="~/CustomInbox.aspx?MessageInstanceViewType=2">
</NavigateToUrlCommand>

```

4. Edit other references as necessary. For example, the Inbox Summary control.

Create a Flowtime Page Without the Top Menu and Side Menu

To create a Flowtime page without the top menu and side menu, add

```
MasterPageFile="~/Layouts/Masters/Empty.master"
```

Example

In this example, we modify the My Tasks page so that only the grid displays for the end user.


```

<%@ Page Title="My Tasks" ResourceTitle="MyTasks" MasterPageFile="~/Layouts/Masters/Empty.Master" MenuId=
"Portal" SubMenuId="MyTasks" SelectedMenuItemId="MyTasks" SelectedSubMenuItemId="MyTasks" Language="C#
" AutoEventWireup="true" CodeBehind="MyTasks.aspx.cs" Inherits="PNMsoft.Sequence.Runtime.Web.UI.Pages.MyT
asks, PNMsoft.Sequence.Runtime.Web.UI, Version=8.0.0.0, Culture=neutral, PublicKeyToken=0a1a1b90c1c5dca1" %
>
<%@ Register TagPrefix="sq" TagName="UserMessagesGridControl"
Src="" %>
<asp:Content ID="Content" ContentPlaceHolderID="Content" runat="server">
<div class="sqpt-container">
<div class="sqpt-grid-control">
<sq:UserMessagesGridControl ID="UserMessagesGridControl" PageSize="20" DisplayVariables="true" ConfigName=
"Portal"
runat="server" MessageInstanceViewItemTypes="TaskOnly" />
</div>
</div>
</asp:Content>

```

Change the Flowtime Website Branding

To change the site's color scheme and branding you should create your own theme.

1. Duplicate the `Sequence8` folder in `C:\Program Files\PNMsoft\Shared Resources\Themes` .
2. Enter a meaningful name for the folder, such as `CustomTheme` .
3. Modify the `flowtimesite.css` file as necessary.
4. Change the `Theme` key in the `web.config` file to your new folder.

```
<add key="PNMsoft.Sequence.Flowtime.Theme" value="MyCustomeTheme"/>
```

Customize Flowtime Grids

You can customize Flowime grids to add a scroll bar and aspects of cell width for Flowtime grids.

Layout

The layout determines the column and table widths. The default column width is 130. You can overwrite the default width in the service configuration file.

Layout Parameters

Parameter	Description
Auto	Column width is set according to the widest unbreakable content in the cells. This option can run slower than the Fixed option, because all content in the grid needs to be read before determining the final layout.
Fixed	<p>This is the default configuration. Horizontal layout only depends on the table's width and the width of the columns, not the contents of the cells. The Fixed layout enables a browser to lay out the table faster than the automatic table layout. The browser can display the table after the first row is read.</p> <p>It is important to be aware of the column widths configured in the global and workflow variables.</p>

Layout - Auto

My Tasks

INBOX: MY OPEN MESSAGES

	From	Subject	Received	Due	Reminder	Workflow Name	WF State
	Sergey Chernyahovsky	Subj about workflow : SergeyTest Instance #195574	9/27/2016				
	Sergey Chernyahovsky	ASD about workflow : SergeyTest Instance #195587	9/27/2016				
	Sergey Chernyahovsky	Task1 in process #195720	10/9/2016	10/10/2016		SergeyWorkflow	●
	Sergey Chernyahovsky	Task1 in process #195803	10/18/2016	10/19/2016		SergeyWorkflow	●
	Sergey Chernyahovsky	Task1 in process #195804	10/18/2016	10/19/2016		SergeyWorkflow	●
	Sergey Chernyahovsky	Task1 in process #195845	10/23/2016	10/24/2016		SergeyWorkflow	●
	Sergey Chernyahovsky	Task1 in process #195948	11/9/2016	11/10/2016		SergeyWorkflow	●
	Marina Torbin	View First Copy in process #2147483769	12/11/2016	12/12/2016		Batch Edit Grid	●
	Marina Torbin	View First in process	12/11/2016	12/12/2016		Batch Edit Grid	●

Layout - Fixed

My Tasks

INBOX: MY OPEN MESSAGES

	From	Subject	Received	Due Date	Reminder Date	Workflow Name	Stage	edited CCN up...	WF State upd
	Sergey Chernyahovsky	Subj about workflow : SergeyTest Instance #195574	9/27/2016						
	Sergey Chernyahovsky	ASD about workflow : SergeyTest Instance #195587	9/27/2016						
	Sergey Chernyahovsky	Task1 in process #195720	10/9/2016	10/10/2016		SergeyWorkflow			●
	Sergey Chernyahovsky	Task1 in process #195803	10/18/2016	10/19/2016		SergeyWorkflow			●
	Sergey Chernyahovsky	Task1 in process #195804	10/18/2016	10/19/2016		SergeyWorkflow			●
	Sergey Chernyahovsky	Task1 in process #195845	10/23/2016	10/24/2016		SergeyWorkflow			●
	Sergey Chernyahovsky	Task1 in process #195948	11/9/2016	11/10/2016		SergeyWorkflow			●
	Marina Torbin	View First Copy in process #2147483769	12/11/2016	12/12/2016		Batch Edit Grid			●
	Marina Torbin	View First in process #2147483769	12/11/2016	12/12/2016		Batch Edit Grid			●
	Marina Torbin	View First Copy in process #2147483770	12/11/2016	12/12/2016		Batch Edit Grid			●
	Marina Torbin	View First in process #2147483770	12/11/2016	12/12/2016		Batch Edit Grid			●
	Marina Torbin	View First Copy in process #2147483790	12/11/2016	12/12/2016		Batch Edit Grid			●
	Marina Torbin	View First in process #2147483790	12/11/2016	12/12/2016		Batch Edit Grid			●
	Marina Torbin	View First Copy in process #2147483799	12/11/2016	12/12/2016		Batch Edit Grid			●
	Marina Torbin	View First in process #2147483799	12/11/2016	12/12/2016		Batch Edit Grid			●
	Marina Torbin	View First Copy in process #2147483800	12/11/2016	12/12/2016		Batch Edit Grid			●

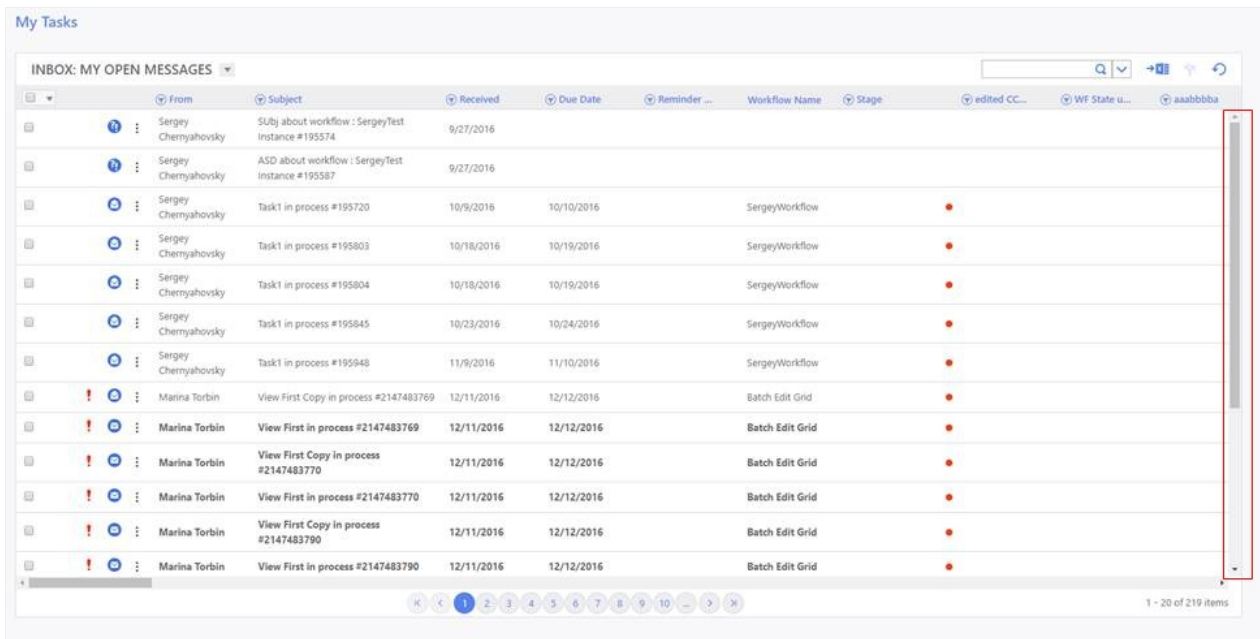
Adaptive

The Adaptive property defines the grid's scroll bar. By default, the Adaptive property is set to *True*, and the layout is set to *Fixed*.

Adaptive Parameters

Parameter	Description
True	Set the Kendo grid native scrollable to <i>True</i> when both the header and table content have their own tables. This can be used for Auto and Fixed layouts.
False	The grid does not have its own scroll bar (but the page is still scrollable).

Adaptive = True



AdaptiveHeight

The AdaptiveHeight property determines the grid's height. By default, the AdaptiveHeight property is set to -1.

AdaptiveHeight Parameters

Value	Description
>0	Custom height is used.
0	The grid determines the height.
-1	The grid auto-resizes with the page. This option is preferable for pages with a single grid.

Customize Column Width

To define local and global variables in the Process page grid, add only the the

`<workflowvariablescolumns>` tag to the grid's `ServiceMetadata.config` file. Make sure you do not add

both the `<workflowvariablescolumns>` and `<globalvariablescolumns>` tags.

Example

```
<workflowvariablescolumns>
  <ColumnTemplates>
  <VariableDataBoundViewColumn
    DataField="Variable2Name" Width="200px"/>
  </ColumnTempates>
</workflowvariablescolumns>
```

V8.4

The Flowtime website enables you to provide Flowtime experience and functionality to end users without having to install SharePoint. It provides the option to add and edit pages quite easily, and because it's a website, there is no need to compile.

Understanding File Structure

After you install Flowtime website, there are several files you can modify that enable you to configure and edit the Flowtime website.

Flowtime website files

- web.config
- ASPX
- Control Template
- Default.config
- ServiceMetada config

web.config files

Allows you to control Flowtime website settings, such as culture, theme, database connection, and so on.

For example, you can set the Flowtime website language by editing the following key:

```
<globalization culture="en-US" uiCulture="en-US" requestEncoding="utf-8" responseEncoding="utf-8"
fileEncoding="utf-8" />
```

ASPX files

These files define the appearance and functionality of each page on the Flowtime website. For example, the `Inbox.aspx` file defines the Inbox page.

Control Template files

The Flowtime website is assembled from pages that host the user controls. For example, `UserMessagesGridControl.ascx` is used to display the User Messages grid.

File Location: `{Site Physical Path}\CONTROLTEMPLATES\Flowtime`

Default.config file

This file defines the top menu links in the Flowtime website. Here you can edit links, set the URL of each link, and add links to new pages.

File Location: C:\Program Files\PNMsoft\Shared Resources\Components\Flowtime\Config\Portal\Commands

ServiceMetadata config files

This set of config files define the way columns display for each grid. For example, the setting for Messages grid columns is defined in the `MessagesServiceMetadata.Portal.Config` file. Files with the word *Portal* in the file name are the files that define grids in the Flowtime website.

File Location: C:\Program Files\PNMsoft\Shared Resources\Components\Flowtime\Config\ServiceMetadata

Edit the Flowtime Website

You can easily add pages, edit pages, and make other appearance and functionality changes to the Flowtime website. We'll cover some of the the most common Flowtime website modifications.

Add a new page

1. Create an ASPX file in the site root folder.
2. Rename the file using a meaningful name.
3. Open and edit the file.

Example

In this example we display the Group Messages Grid control and Form Viewer control on the new page `MyPage.aspx`. The Form Viewer control enables you to add any form view to a page without referencing a specific process. The sections we added are highlighted.

For the controls to work, you need to add a `<Register>` tag for each control on the page that you add. You can find these links in the Flowtime Application in the IIS.

IMPORTANT: If you need to add your own code behind for the page, you must inherit from the `PortalBasePage` class from `PNMsoft.Sequence.Runtime.Web.UI.dll`.

```
<%@ Page Title="" Language="C#" CodeBehind="Default.aspx.cs"
Inherits="PNMsoft.Sequence.Runtime.Web.UI.Pages.Default" AutoEventWireup="true"
%>
<%@ Register TagPrefix="sq" TagName="GroupMessagesGridControl"
Src="" %>
<%@ Register TagPrefix="sq" TagName="FormViewerControl"
Src="" %>
<asp:Content ID="Content" ContentPlaceHolderID="content" runat="server">
  <div class="sqpt-container">

    <div class="sqpt-grid-control">
      <sq:GroupMessagesGridControl ID="GroupMessagesGridControl"
      PageSize="5" DisplayVariables="false"
      runat="server" />
    </div>

    <sq:FormViewerControl ID="ReportsView"
    FormVirtualPath="~/SequenceForms/wf/Test/Test/Form1/DefaultView.ascx"
    runat="server" />
  </div>
</asp:Content>
```

Add a Page that Includes a Dashboard

1. Access the ASPX file that you created in the root folder.
2. Add the following AnalyticsDashboardControl declaration.

```
<%@ Register TagPrefix="sq" TagName="AnalyticsDashboardControl"
Src="" %>
```

3. Add the following markup to the page.

```
<asp:Content ID="Content" ContentPlaceHolderID="Content" runat="server">
  <div class="sqpt-container">
    <sq:AnalyticsDashboardControl runat="server" ID="AnalyticsDashboardControl" Transport="flowtime" ThemeName="Default" Mode="Auto"></sq:AnalyticsDashboardControl>
  </div>
</asp:Content>
```

4. Configure the following properties.

Property	Description
DataStreamName	name of the dashboard suite you are using.
Layout	name of the dashboard from the dashboard suite that you want to display
Transport	by default, the Transport is <i>flowtime</i> . Only advanced users should modify this property.

5. (Optional) Configure the follow properties as necessary.

Property	Description
ThemeName	by default, ThemeName is <i>Default</i> . The other valid value is <i>Compact</i> .
Mode	by default, Mode is <i>Auto</i> . The other valid values are <i>Fixed</i> and <i>Fluid</i> . This affects the dashboard layout, whether it is fixed, fluid, or set automatically.

6. (Optional) Advanced users can configure the following properties as necessary.

Table Title

Property	Description
Parameters	can include different context parameters, such as <i>SolutionId</i> of a <i>HotOperations</i> solution, and so on. For this to work, users should inherit the code-behind from specific classes.
DataStreamsURL	points to the data stream's Web service.

The final ASPX file will contain the following markup.

```
<asp:Content ID="Content" ContentPlaceHolderID="Content" runat="server">
  <div class="sqpt-container">
    <sq:AnalyticsDashboardControl runat="server" ID="AnalyticsDashboardControl"
Transport="flowtime" ThemeName="Default" Mode="Auto"></sq:AnalyticsDashboardControl>
  </div>
</asp:Content>
```

Add a Link to the Banner

1. Open the `Default.config` file.
2. Copy a `<MenuItem>` section, and paste it under the existing `<MenuItem>` section.

```
<MenuItem
  Id="Inbox"
  Command="Inbox"
  Text="{ $resources.Inbox}"
  Tooltip="{ $resources.Inbox}">
</MenuItem>
```

3. Edit the values in the `<MenuItem>` section.

```
<MenuItem
  Id="MyPage"
  Command="MyPage"
  Text="My Page"
  Tooltip="MY Page">
</MenuItem>
```

4. Copy a `NavigateToUrlCommand` section and paste it below the existing `NavigateToUrlCommand` section. This section determines the URL of the link.

```
<NavigateToUrlCommand
  Id="Inbox"
  Url="~/Inbox.aspx?MessageInstanceViewType=2">
</NavigateToUrlCommand>
```

5. Edit the values in the `NavigateToUrlCommand` section with the following information.

```
<NavigateToUrlCommand
  Id="MyPage"
  Url="~/MyPage.aspx">
</NavigateToUrlCommand>
```

6. Perform IIS Reset and refresh the homepage.

Edit an Existing Page

1. Make a copy of the page you want to edit in the root folder. For example, copy the `Inbox.aspx` file to the `MyPage.aspx` file.
2. Edit the file as necessary. In the example below, we added a new user control to the `Inbox.aspx` file. The added sections are highlighted.

```

<%@ Page Title="" Language="C#" MenuId="Portal" SelectedMenuItemId="MyPage" AutoEventWireup="false
"
CodeBehind="Inbox.aspx.cs"
Inherits="PNMsoft.Sequence.Runtime.Web.UI.Pages.Inbox" %>
<%@ Register TagPrefix="sq" TagName="InboxSummaryControl"
Src="" %>
<%@ Register TagPrefix="sq" TagName="NewProcessControl"
Src="" %>
<%@ Register TagPrefix="sq" TagName="ProcessesIOwnControl"
Src="" %>
<%@ Register TagPrefix="sq" TagName="UserMessagesGridControl"
Src="" %>
<%@ Register TagPrefix="sq" TagName="GroupMessagesGridControl"
Src="" %>
<asp:Content ID="Content" ContentPlaceHolderID="content" runat="server">
<div class="sqpt-container">
<div class="sqpt-summary-controls">
<div>
<sq:NewProcessControl ID="NewProcessControl"
runat="server" />
</div>
<div>
<sq:InboxSummaryControl ID="InboxSummaryControl" InboxPage="Inbox.aspx"
runat="server" />
</div>
<div>
<sq:ProcessesIOwnControl ID="ProcessesIOwnControl" ProcessesIOwnPage="OpenProcesses.aspx" runat
="server" />
</div>
</div>
<div class="sqpt-grid-control">
<sq:UserMessagesGridControl ID="UserMessagesGridControl"
PageSize="30" DisplayVariables="true" ConfigName="Portal"
runat="server" />
</div>
<div class="sqpt-grid-control">
<sq:GroupMessagesGridControl ID="GroupMessagesGridControl"
PageSize="5" DisplayVariables="false"
runat="server" />
</div>
</div>

```

3. Change the page reference in the `Default.config` file, which is located at `C:\Program Files\PNMsoft\Shared Resources\Components\Flowtime\Config\Portal\Commands`.

```

<NavigateToUrlCommand
Id="Inbox"
Url="~/CustomInbox.aspx?MessageInstanceViewType=2">
</NavigateToUrlCommand>

```

4. Edit other references as necessary. For example, the Inbox Summary control.

Change the Flowtime Website Branding

To change the site's color scheme and branding you should create your own theme.

1. Duplicate the `Sequence8` folder in `C:\Program Files\PNMsoft\Shared Resources\Themes`.
2. Enter a meaningful name for the folder, such as `CustomTheme`.
3. Modify the `flowtimesite.css` file as necessary.
4. Change the `Theme` key in the `web.config` file to your new folder.


```
<add key="PNMsoft.Sequence.Flowtime.Theme" value="MyCustomeTheme"/>
```

Customize Flowtime Grids

You can customize Flowime grids to add a scroll bar and aspects of cell width for Flowtime grids.

Layout

The layout determines the column and table widths. The default column width is 130. You can overwrite the default width in the service configuration file.

Layout Parameters

Parameter	Description
Auto	Column width is set according to the widest unbreakable content in the cells. This option can run slower than the Fixed option, because all content in the grid needs to be read before determining the final layout.
Fixed	<p>This is the default configuration. Horizontal layout only depends on the table's width and the width of the columns, not the contents of the cells. The Fixed layout enables a browser to lay out the table faster than the automatic table layout. The browser can display the table after the first row is read.</p> <p>It is important to be aware of the column widths configured in the global and workflow variables.</p>

Layout - Auto

My Tasks

INBOX: MY OPEN MESSAGES

	From	Subject	Received	Due	Reminder	Workflow Name	WF State
	Sergey Chernyahovsky	Subj about workflow : SergeyTest Instance #195574	9/27/2016				
	Sergey Chernyahovsky	ASD about workflow : SergeyTest Instance #195587	9/27/2016				
	Sergey Chernyahovsky	Task1 in process #195720	10/9/2016	10/10/2016		SergeyWorkflow	●
	Sergey Chernyahovsky	Task1 in process #195803	10/18/2016	10/19/2016		SergeyWorkflow	●
	Sergey Chernyahovsky	Task1 in process #195804	10/18/2016	10/19/2016		SergeyWorkflow	●
	Sergey Chernyahovsky	Task1 in process #195845	10/23/2016	10/24/2016		SergeyWorkflow	●
	Sergey Chernyahovsky	Task1 in process #195948	11/9/2016	11/10/2016		SergeyWorkflow	●
	Marina Torbin	View First Copy in process #2147483769	12/11/2016	12/12/2016		Batch Edit Grid	●
	Marina Torbin	View First in process	12/11/2016	12/12/2016		Batch Edit Grid	●

Layout - Fixed

My Tasks

INBOX: MY OPEN MESSAGES

	From	Subject	Received	Due Date	Reminder Date	Workflow Name	Stage	edited CCN up...	WF State upd
	Sergey Chernyahovsky	Subj about workflow : SergeyTest Instance #195574	9/27/2016						
	Sergey Chernyahovsky	ASD about workflow : SergeyTest Instance #195587	9/27/2016						
	Sergey Chernyahovsky	Task1 in process #195720	10/9/2016	10/10/2016		SergeyWorkflow			●
	Sergey Chernyahovsky	Task1 in process #195803	10/18/2016	10/19/2016		SergeyWorkflow			●
	Sergey Chernyahovsky	Task1 in process #195804	10/18/2016	10/19/2016		SergeyWorkflow			●
	Sergey Chernyahovsky	Task1 in process #195845	10/23/2016	10/24/2016		SergeyWorkflow			●
	Sergey Chernyahovsky	Task1 in process #195948	11/9/2016	11/10/2016		SergeyWorkflow			●
	Marina Torbin	View First Copy in process #2147483769	12/11/2016	12/12/2016		Batch Edit Grid			●
	Marina Torbin	View First in process #2147483769	12/11/2016	12/12/2016		Batch Edit Grid			●
	Marina Torbin	View First Copy in process #2147483770	12/11/2016	12/12/2016		Batch Edit Grid			●
	Marina Torbin	View First in process #2147483770	12/11/2016	12/12/2016		Batch Edit Grid			●
	Marina Torbin	View First Copy in process #2147483790	12/11/2016	12/12/2016		Batch Edit Grid			●
	Marina Torbin	View First in process #2147483790	12/11/2016	12/12/2016		Batch Edit Grid			●
	Marina Torbin	View First Copy in process #2147483799	12/11/2016	12/12/2016		Batch Edit Grid			●
	Marina Torbin	View First in process #2147483799	12/11/2016	12/12/2016		Batch Edit Grid			●
	Marina Torbin	View First Copy in process #2147483800	12/11/2016	12/12/2016		Batch Edit Grid			●

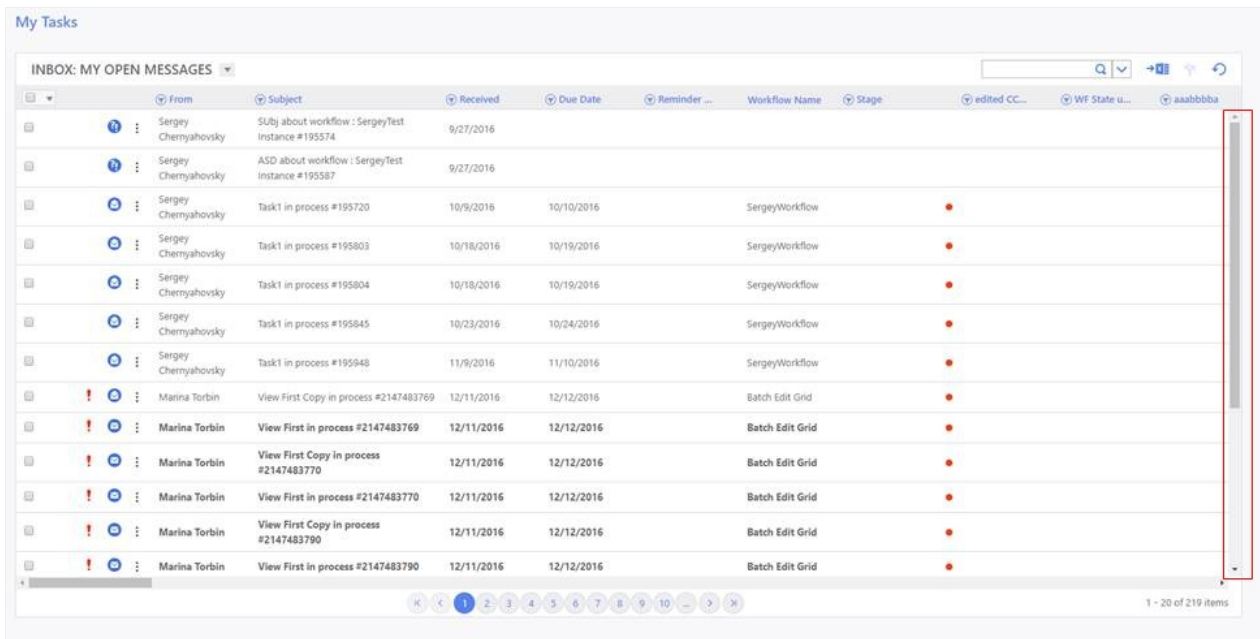
Adaptive

The Adaptive property defines the grid's scroll bar. By default, the Adaptive property is set to *True*, and the layout is set to *Fixed*.

Adaptive Parameters

Parameter	Description
True	Set the Kendo grid native scrollable to <i>True</i> when both the header and table content have their own tables. This can be used for Auto and Fixed layouts.
False	The grid does not have its own scroll bar (but the page is still scrollable).

Adaptive = True



AdaptiveHeight

The AdaptiveHeight property determines the grid's height. By default, the AdaptiveHeight property is set to -1.

AdaptiveHeight Parameters

Value	Description
>0	Custom height is used.
0	The grid determines the height.
-1	The grid auto-resizes with the page. This option is preferable for pages with a single grid.

Customize Column Width

To define local and global variables in the Process page grid, add only the the

`<workflowvariablescolumns>` tag to the grid's `ServiceMetadata.config` file. Make sure you do not add

both the `<workflowvariablescolumns>` and `<globalvariablescolumns>` tags.

Example

```
<workflowvariablescolumns>
  <ColumnTemplates>
  <VariableDataBoundViewColumn
    DataField="Variable2Name" Width="200px"/>
  </ColumnTempates>
</workflowvariablescolumns>
```

V8.3

The Flowtime website enables you to provide user experience and functionality to end users without having to install SharePoint. It provides the option to add and edit pages quite easily, and because it's a website, there is no need to compile.

Understanding File Structure

After you install Flowtime website, there are several files you can modify that enable you to configure and edit the Flowtime website.

Flowtime website files

- web.config
- ASPX
- Control Template
- Default.config
- ServiceMetada config

web.config files

Allows you to control Flowtime website settings, such as culture, theme, database connection, and so on.

For example, you can set the Flowtime website language by editing the following key:

```
<globalization culture="en-US" uiCulture="en-US" requestEncoding="utf-8" responseEncoding="utf-8"
fileEncoding="utf-8" />
```

ASPX files

These files define the appearance and functionality of each page on the Flowtime website. For example, the `Inbox.aspx` file defines the Inbox page.

Control Template files

The Flowtime website is assembled from pages that host the user controls. For example, `UserMessagesGridControl.ascx` is used to display the User Messages grid.

File Location: `{Site Physical Path}\CONTROLTEMPLATES\Flowtime`

Default.config file

This file defines the top menu links in the Flowtime website. Here you can edit links, set the URL of each link, and add links to new pages.

File Location: C:\Program Files\PNMsoft\Shared Resources\Components\Flowtime\Config\Portal\Commands

ServiceMetadata config files

This set of config files define the way columns display for each grid. For example, the setting for Messages grid columns is defined in the `MessagesServiceMetadata.Portal.Config` file. Files with the word *Portal* in the file name are the files that define grids in the Flowtime website.

File Location: C:\Program Files\PNMsoft\Shared Resources\Components\Flowtime\Config\ServiceMetadata

Edit the Flowtime Website

You can easily add pages, edit pages, and make other appearance and functionality changes to the Flowtime website. We'll cover some of the the most common Flowtime website modifications.

Add a new page

1. Create an ASPX file in the site root folder.
2. Rename the file using a meaningful name.
3. Open and edit the file.

Example

In this example we display the Group Messages Grid control and Form Viewer control on the new page `MyPage.aspx`. The Form Viewer control enables you to add any form view to a page without referencing a specific process. The sections we added are highlighted.

For the controls to work, you need to add a `<Register>` tag for each control on the page that you add. You can find these links in the Flowtime Application in the IIS.

IMPORTANT: If you need to add your own code behind for the page, you must inherit from the `PortalBasePage` class from `PNMsoft.Sequence.Runtime.Web.UI.dll`.

```
<%@ Page Title="" Language="C#" CodeBehind="Default.aspx.cs"
Inherits="PNMsoft.Sequence.Runtime.Web.UI.Pages.Default" AutoEventWireup="true"
%>
<%@ Register TagPrefix="sq" TagName="GroupMessagesGridControl"
Src="" %>
<%@ Register TagPrefix="sq" TagName="FormViewerControl"
Src="" %>
<asp:Content ID="Content" ContentPlaceHolderID="content" runat="server">
  <div class="sqpt-container">

    <div class="sqpt-grid-control">
      <sq:GroupMessagesGridControl ID="GroupMessagesGridControl"
      PageSize="5" DisplayVariables="false"
      runat="server" />
    </div>

    <sq:FormViewerControl ID="ReportsView"
    FormVirtualPath="~/SequenceForms/wf/Test/Test/Form1/DefaultView.ascx"
    runat="server" />
  </div>
</asp:Content>
```

Add a Page that Includes a Dashboard

1. Access the ASPX file that you created in the root folder.
2. Add the following AnalyticsDashboardControl declaration.

```
<%@ Register TagPrefix="sq" TagName="AnalyticsDashboardControl"
Src="" %>
```

3. Add the following markup to the page.

```
<asp:Content ID="Content" ContentPlaceHolderID="Content" runat="server">
  <div class="sqpt-container">
    <sq:AnalyticsDashboardControl runat="server" ID="AnalyticsDashboardControl" Transport="flowtime" ThemeName="Default" Mode="Auto"></sq:AnalyticsDashboardControl>
  </div>
</asp:Content>
```

4. Configure the following properties.

Property	Description
DataStreamName	name of the dashboard suite you are using.
Layout	name of the dashboard from the dashboard suite that you want to display
Transport	by default, the Transport is <i>flowtime</i> . Only advanced users should modify this property.

5. (Optional) Configure the follow properties as necessary.

Property	Description
ThemeName	by default, ThemeName is <i>Default</i> . The other valid value is <i>Compact</i> .
Mode	by default, Mode is <i>Auto</i> . The other valid values are <i>Fixed</i> and <i>Fluid</i> . This affects the dashboard layout, whether it is fixed, fluid, or set automatically.

6. (Optional) Advanced users can configure the following properties as necessary.

Table Title

Property	Description
Parameters	can include different context parameters, such as <i>SolutionId</i> of a HotOperations solution, and so on. For this to work, users should inherit the code-behind from specific classes.
DataStreamsURL	points to the data stream's Web service.

The final ASPX file will contain the following markup.

```
<asp:Content ID="Content" ContentPlaceHolderID="Content" runat="server">
  <div class="sqpt-container">
    <sq:AnalyticsDashboardControl runat="server" ID="AnalyticsDashboardControl"
Transport="flowtime" ThemeName="Default" Mode="Auto"></sq:AnalyticsDashboardControl>
  </div>
</asp:Content>
```

Add a Page Without the Top Menu and Side Menu

You can customize any Flowtime page so that the header and side bar do not display on the page.

Add a Link to the Top Banner

1. Open the `Default.config` file.
2. Copy a `<MenuItem>` section, and paste it under the existing `<MenuItem>` section.

```
<MenuItem
  Id="Inbox"
  Command="Inbox"
  Text="{ $resources.Inbox}"
  Tooltip="{ $resources.Inbox}">
</MenuItem>
```

3. Edit the values in the `<MenuItem>` section.

```
<MenuItem
  Id="MyPage"
  Command="MyPage"
  Text="My Page"
  Tooltip="MY Page">
</MenuItem>
```

4. Copy a `NavigateToUrlCommand` section and paste it below the existing `NavigateToUrlCommand` section. This section determines the URL of the link.

```
<NavigateToUrlCommand
  Id="Inbox"
  Url="~/Inbox.aspx?MessageInstanceViewType=2">
</NavigateToUrlCommand>
```

5. Edit the values in the `NavigateToUrlCommand` section with the following information.

```
<NavigateToUrlCommand
  Id="MyPage"
  Url="~/MyPage.aspx">
</NavigateToUrlCommand>
```

6. Perform IIS Reset and refresh the homepage.

Edit an Existing Page

1. Make a copy of the page you want to edit in the root folder. For example, copy the `Inbox.aspx` file to the `MyPage.aspx` file.
2. Edit the file as necessary. In the example below, we added a new user control to the `Inbox.aspx` file. The added sections are highlighted.

```

<%@ Page Title="" Language="C#" MenuId="Portal" SelectedMenuItemId="MyPage" AutoEventWireup="false
"
CodeBehind="Inbox.aspx.cs"
Inherits="PNMsoft.Sequence.Runtime.Web.UI.Pages.Inbox" %>
<%@ Register TagPrefix="sq" TagName="InboxSummaryControl"
Src="" %>
<%@ Register TagPrefix="sq" TagName="NewProcessControl"
Src="" %>
<%@ Register TagPrefix="sq" TagName="ProcessesIOwnControl"
Src="" %>
<%@ Register TagPrefix="sq" TagName="UserMessagesGridControl"
Src="" %>
<%@ Register TagPrefix="sq" TagName="GroupMessagesGridControl"
Src="" %>
<asp:Content ID="Content" ContentPlaceHolderID="content" runat="server">
<div class="sqpt-container">
<div class="sqpt-summary-controls">
<div>
<sq:NewProcessControl ID="NewProcessControl"
runat="server" />
</div>
<div>
<sq:InboxSummaryControl ID="InboxSummaryControl" InboxPage="Inbox.aspx"
runat="server" />
</div>
<div>
<sq:ProcessesIOwnControl ID="ProcessesIOwnControl" ProcessesIOwnPage="OpenProcesses.aspx" runat
="server" />
</div>
</div>
<div class="sqpt-grid-control">
<sq:UserMessagesGridControl ID="UserMessagesGridControl"
PageSize="30" DisplayVariables="true" ConfigName="Portal"
runat="server" />
</div>
<div class="sqpt-grid-control">
<sq:GroupMessagesGridControl ID="GroupMessagesGridControl"
PageSize="5" DisplayVariables="false"
runat="server" />
</div>
</div>
</div>

```

3. Change the page reference in the `Default.config` file, which is located at `C:\Program Files\PNMsoft\Shared Resources\Components\Flowtime\Config\Portal\Commands`.

```

<NavigateToUrlCommand
Id="Inbox"
Url="~/CustomInbox.aspx?MessageInstanceViewType=2">
</NavigateToUrlCommand>

```

4. Edit other references as necessary. For example, the Inbox Summary control.

Change the Flowtime Website Branding

To change the site's color scheme and branding you should create your own theme.

1. Duplicate the `Sequence8` folder in `C:\Program Files\PNMsoft\Shared Resources\Themes`.
2. Enter a meaningful name for the folder, such as `CustomTheme`.
3. Modify the `flowtimesite.css` file as necessary.
4. Change the `Theme` key in the `web.config` file to your new folder.


```
<add key="PNMsoft.Sequence.Flowtime.Theme" value="MyCustomeTheme"/>
```

For more details, see the [Flowtime Website Configuration guide](#).

V8.2

The Flowtime Website Configuration guide describes how to configure Flowtime for Cora SeQuence 8.2.

Click [here](#) to open the guide.

For information on installing Flowtime on Cora SeQuence V8.2 and later, see the [Cora SeQuence Installation Guide](#).

V8.1

The *Flowtime Website Installation and Configuration* guide describes how to install and configure the Flowtime website for Cora SeQuence 8.1.

Click [here](#) to open the guide.