

Widget SDK

1	Java Widget Example	2
1.1	Introduction	2
1.2	Java API Workspace preparation	2
1.3	Creating project plugin	2
1.3.1	Import of the example	2
1.4	Deployment to Installed Polarion	2
1.5	Execution from Workspace	2
1.6	Configuration	2
2	Velocity Widget Example	2
2.1	Introduction	2
2.2	Deployment	3
2.2.1	Import of the example	3
2.3	Tags	3
3	Velocity Macro Example	3
3.1	Introduction	3
3.2	Deployment	3
3.2.1	Import of the example	3
3.2.2	Usage	3
4	Customize Existing Widget in Script Widget Example	3
4.1	Introduction	3
4.2	Implementation	4
5	Extend Velocity Context Example	4
5.1	Introduction	4
5.2	Java API Workspace preparation	5
5.3	Creating project plugin	5
5.3.1	Import of the example	5
5.4	Deployment to Installed Polarion	5
5.5	Execution from Workspace	5
5.6	Configuration	5
5.6.1	Example usage of Work Item Util	5
6	Small Examples	5
6.1	Create a Custom Highcharts Chart	5
6.2	Setting Raw Attributes for a Highcharts Chart	6
6.3	Render a Set of Objects as a Table	6
6.4	Widget Parameter for Selecting Fields	7
6.5	Get Historical Object by Date	7
6.6	How to Use Page Parameters in Velocity and Queries	8
6.6.1	String Parameters	8
6.6.2	Integer Parameters	9
6.6.3	Boolean Parameters	9
6.6.4	Enumeration Parameters	9
6.6.5	Custom Enumeration Parameters	9
6.6.6	Date Parameters	9
6.6.7	Usage in "Lucene + Velocity" and "SQL + Velocity" Queries	10
7	Page Script	10
7.1	Page Context	10
7.1.1	Page Context Example	10
7.2	Scripted Page Parameters	11
7.2.1	Scripted Page Parameters example	12
8	Dependencies between widget parameters	12
8.1	Example	12
8.1.1	Java widget	12
8.1.2	Velocity widget	12
9	Actions in widgets	13
9.1	How it works	13
9.2	Example	13
10	Velocity Context Reference	13
10.1	Available Everywhere	13
10.1.1	Polarion Objects	13
10.1.2	Polarion Services (from Polarion Open Java API)	14
10.1.3	Velocity Generic Tools	14
10.2	Available in Velocity Widgets and Page Script	14
10.2.1	Velocity Widget Objects	14
11	Examples of using ParameterFactory in Velocity	15
11.1	Adding scripted page parameters in Page Script	15

11.2 Adding widget parameters in parameters.vm	15
12 Predefined Velocity Macros	15
13 Enabling the Work Item Properties Sidebar in a Custom Widget	16
13.1 What HTML attributes are configured on a Work Item so that the Work Items Sidebar is displayed	16
13.2 Showing Different Fields for Different Groups of Work Items	16
14 Customizing the Kanban Board	16
14.1 Examples	17
14.1.1 How to change the card color	17
14.1.2 How to add a Work Item's Tree Structure link	18
15 Frequently Asked Questions	18

1 Java Widget Example

1.1 Introduction

In this example we will show how to create a custom Java Widget for pages.

What is possible to extend

- By developing your own Widgets you can extend your Widget library in Pages.

What will be shown in the example

- How to create a custom Widget. We will implement a *Work Record Report* widget which calculates work records from items defined by its parent and link role.

1.2 Java API Workspace preparation

See section *Workspace preparation* in main Polarion SDK Guide.

1.3 Creating project plugin

The best way is to import the provided example, which contains all dependencies you need for custom plugin creation.

1.3.1 Import of the example

Info: You must ensure that your plugin is compiled against your Polarion version. This example contains a precompiled jar plugin. You can remove it before you start developing your plugin based on this example. Eclipse ensures that the new jar plugin will be created against your source code and Polarion version.

1. Select **File > Import...**
2. In the dialog that appears, select **Existing Project into Workspace** in the **General** section and press the **Next** button.
3. Pressing the **Browse** button, and select the directory of examples (usually in `C:\Polarion\polarion\SDK\examples\`). Submit it.
4. Select `com.polarion.example.widget` and press **Finish**.

1.4 Deployment to Installed Polarion

See section *Deployment to Installed Polarion* in main Polarion SDK guide.

1.5 Execution from Workspace

See section *Execution from Workspace* in main Polarion SDK guide.

1.6 Configuration

After successful deployment of the plug-in into Polarion, you can start using the Work Record Widget in Pages.

Info: Custom widget plugin is deployed using HiveMind. See `/src/META-INF/hivemodule.xml` for more details. Deployment configuration is located at `/com.polarion.alm.ui.jar/META-INF/richtpages-hivemodule.xml`.

2 Velocity Widget Example

2.1 Introduction

In this example we will show how to create custom Velocity based Widget for Pages.

What is possible to extend

- By developing your own Widgets you can extend your Widget library in pages.

What will be shown in the example

- How to create a custom Widget. We will implement a *Work Record Report* widget that calculates work records from items defined by its parent and link role.

2.2 Deployment

2.2.1 Import of the example

1. Use your favorite SVN client.
2. Commit contents of `com.polarion.example.velocitywidget` (usually in `C:\Polarion\polarion\SDK\examples\`) to repository location `.polarion/pages/widgets`.
3. Now you can use the new Widget from the Widgets Sidebar in the Page Designer (i.e. a Page in edit mode).

Info: Velocity Widgets can be deployed in the repository in global and project scopes. Widgets deployed in the global scope will be visible to all projects. If there are two Widgets with the same ID deployed in both global and project scopes, then the Widget in the project scope will be used.

2.3 Tags

Tags (or categories) of the widget, which are specified as comma separated list in the `tags` property in `widget.properties` file, can be any String or localization keys. For localization keys of the standard tags see javadoc of `com.polarion.alm.shared.api.model.rp.widget.RichPageWidget.getTags(SharedContext)`

3 Velocity Macro Example

3.1 Introduction

In this example we will show how to deploy a custom macro for Pages.

What is possible to extend

- By developing your own macros you can extend your macro library in Pages.

What will be shown in the example

- How to create custom macro. We will implement a custom Info box to display information.

3.2 Deployment

3.2.1 Import of the example

1. Use your favorite SVN client.
2. Commit contents of `com.polarion.example.velocitymacro` (usually in `C:\Polarion\polarion\SDK\examples\`) to repository location `.polarion/pages/scripts/velocity`.
3. Now you can use new macro from the Script - Block Widget and Script - Inline Widget.

3.2.2 Usage

1. Place Script - Block Widget or Script - Inline Widget to your Page from Widgets Sidebar
2. Import macros using `#import("macros.vm")`.
3. Use infoBox macro like `#infoBox("your message")`.

Info: The included file is first searched in the project scope. If it does not exist there, then it is taken from global scope.

4 Customize Existing Widget in Script Widget Example

4.1 Introduction

In this example we will show how to customize an existing Widget in scripts using **Copy Widget Source Code** operation.

What will be shown in the example

- How to customize existing **Multi Line Trend Chart** Widget - we will display number of work items separated by category in time axis.

4.2 Implementation

1. Insert a **Multi Line Trend Chart** Widget into your Page.
2. Change title of the Widget and press Apply. This is necessary so that all parameters are copied into the Widget source code. We need them for next steps.
3. Click on the Operation menu button in the Widget header and select menu item **Copy Widget Source Code**.
4. Insert a **Script Block** Widget and paste the copied source code into its *Script* parameter in the opened Parameters sidebar. You will see the basic Widget definition with pre-filled parameters.
5. Modify the source code to look like the code shown below. (The only changes should be the first line with Velocity assignment and content of the `<sub id="elements"> element`):

```
#set($projectId = "PROJECT_ID")
<div class="polarion-rp-widget-part" data-widget="com.polarion.multiSetTrendChart">
  <span class="polarion-rp-widget-parameters">
    <sub id="title">Work Items count separated by category</sub>
    <sub id="data">
      <sub id="elements">
        #set($categories =
$transaction.categories().search().query("project.id:$projectId"))
        #foreach($category in $categories)
          <sub>
            <sub id="prototype">WorkItem</sub>
            <sub id="queryType">lucene</sub>
            <sub id="luceneQuery">categories.id:$category.fields.id.get</sub>
            <sub id="children">
              <sub id="name">$category.fields.name.get</sub><sub id="color"></sub>
            </sub>
          </sub>
        #end
      </sub>
    </sub>
  </span>
  <sub id="timeAxis">
    <sub id="from">
      <sub id="relative">-30</sub>
    </sub>
    <sub id="to">
      <sub id="relative">0</sub>
    </sub>
    <sub id="scaleType">week</sub>
    <sub id="baseDate">1</sub>
  </sub>
</div>
```

6. Change variable `projectId` to point to your chosen project.
7. Press Apply.

Info: There are two kinds of Script widgets - Inline and Block. The Inline one can be used inside paragraph, as it does not break the text. Block widgets are used as separate components.

5 Extend Velocity Context Example

5.1 Introduction

In this example we will show how to deploy and use your own HiveMind services or Java instances in scripts, Widgets and macros.

What is possible to extend

- By developing your own services and objects you can extend the Velocity context used in Pages, Widgets, macros and parameters which allow use of Velocity.

What will be shown in the example

- How to extend the Velocity context. We will implement *Work Item Util* contributed as a Java instance, which will count time spent on a given Work Item. Next implementation will be *Repository Util* contributed as a HiveMind service, which provides a method for getting

the last revision in the repository, and also provides *IRepositoryService*.

5.2 Java API Workspace preparation

See section *Workspace preparation* in main Polarion SDK Guide.

5.3 Creating project plugin

The best way is to import the provided example, which contains all dependencies you need for custom plugin creation.

5.3.1 Import of the example

Info: You must ensure that your plugin is compiled against your Polarion version. This example contains precompiled jar plugin. You can remove it before you start developing your plugin based on this example. Eclipse ensures that the new jar plugin will be created against your source code and Polarion version.

1. Select **File > Import...**
2. In the dialog that appears, select **Existing Project into Workspace** in the **General** section and press the **Next** button.
3. Pressing the **Browse** button, then select the directory of examples (usually in `C:\Polarion\polarion\SDK\examples\`). Submit it.
4. Select `com.polarion.example.velocitycontext` and press **Finish**.

5.4 Deployment to Installed Polarion

See section *Deployment to Installed Polarion* in main Polarion SDK guide.

5.5 Execution from Workspace

See section *Execution from Workspace* in main Polarion SDK guide.

5.6 Configuration

After successful deployment of the plug-in into Polarion, you can start using *Work Item Util* and *Repository Util* within your Velocity code in Pages:

- `$workItemUtil` - Work Item Util
- `$repositoryUtil` - Repository Util

5.6.1 Example usage of Work Item Util

Create page with e.g. a Script Block Widget. As content of the script, use:

```
$workItemUtil.printTimeSpent($transaction.workItems().getBy().ids("PROJECT_ID",  
"WORKITEM_ID"))
```

Replace `PROJECT_ID` and `WORKITEM_ID` with identifications of project and Work Item which has work records.

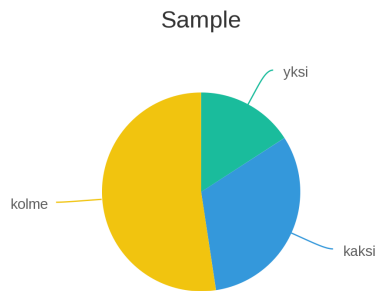
Info: Velocity Context plugin is deployed using HiveMind. See `/src/META-INF/hivemodule.xml` for more details.

6 Small Examples

6.1 Create a Custom Highcharts Chart

This example shows how to use the Script Block widget to generate a custom Highcharts chart using Velocity and the Polarion Rendering API.

```
#set($html = $renderingContext.createHtmlFragmentBuilder())  
  
#set($chart = $renderingContext.createChartBuilder())  
#set($pie = $chart.pie().name("Number of demands"))  
  
$!pie.addValue("yksi", 1).null  
$!pie.addValue("kaksi", 2).null  
$!pie.addValue("kolme", $math.toDouble("3.3")).null  
  
$!chart.build().title().text("Sample").null  
$!chart.build().plotOptions().pie().addRawAttribute("tooltip", "{pointFormat: '<b>{point.y}</b>'}").null  
  
$!chart.build().render($html, 300, $renderingContext.columnWidth).null  
  
$html
```



6.2 Setting Raw Attributes for a Highcharts Chart

This example shows how to set raw attributes for these charts.

```
#set($html = $renderingContext.createHtmlFragmentBuilder())
#set($builder = $renderingContext.createChartBuilder().build())

$!builder.title().text("Chart").null
$!builder.chart().type().line().null

$!builder.addRowAttribute("noData","{style: {fontWeight: 'bold', fontSize: '15px', color: '#FF0000'}}").null

$!builder.render($html, 300, $renderingContext.columnWidth).null

$html
```

Chart

No data to display

6.3 Render a Set of Objects as a Table

The following code examples show how to use the Script Block widget to render a table that displays properties of objects. The object set is defined by a query.

EXAMPLE 1:

```
<table class="polarion-rpw-table-content">
  <tbody>
    <tr class="polarion-rpw-table-header-row">
      <th>Item</th>
      <th>Status</th>
      <th>Parent</th>
    </tr>
    #foreach($w in $transaction.workItems.search.query("project.id:drivepilot"))
      <tr class="polarion-rpw-table-content-row">
        <td>$w.render.withLinks.withTitle</td>
        <td>$w.fields.status.render</td>
        <td>
          #foreach($l in $w.fields.linkedWorkItems.direct)
            #if($l.fields.role.get.id == "parent")
              $l.fields.workItem.render.withTitle.withLinks
            #end
          #end
        </td>
      </tr>
    #end
  </tbody>
</table>
```

Result of the above:

Item	Status	Parent
SDK-2 - Child Requirement	Open	SDK-1 - Parent Requirement
SDK-1 - Parent Requirement	Open	

EXAMPLE 2:

```
<table class="polarion-rpw-table-content">
  <tbody>
    <tr class="polarion-rpw-table-header-row">
      <th>Plan #</th>
      <th># of Work Items at Start</th>
      <th># of WIs at End</th>
    </tr>

    #set($counter=1)

    #foreach($plan in $transaction.plans().search().limit(19))
      #if($plan.fields().startedOn().get() && $plan.fields().finishedOn().get() &&
      $plan.fields().name().get().contains("Team"))
        #set($revisionStarted =
$transaction.revisions().getBy().dateOrAfter($plan.fields().startedOn().get()).fields().id().get())
        #set($planAtStart =
$transaction.plans().getBy().revision($revisionStarted).path($plan.getReference().toPath()))
        #set($revisionFinished =
$transaction.revisions().getBy().dateOrAfter($plan.fields().finishedOn().get()).fields().id().get())
        #set($planAtEnd =
$transaction.plans().getBy().revision($revisionFinished).path($plan.getReference().toPath()))
        #if(!$planAtStart.isUnresolvable() && !$planAtEnd.isUnresolvable())

          <tr class="polarion-rpw-table-content-row">
            <td>$counter</td>
            <td>$planAtStart.items().size()</td>
            <td>$planAtEnd.items().size()</td>
          </tr>

          #set($counter=$math.add($counter, 1))
        #end
      #end
    #end

  </tbody>
</table>
```

6.4 Widget Parameter for Selecting Fields

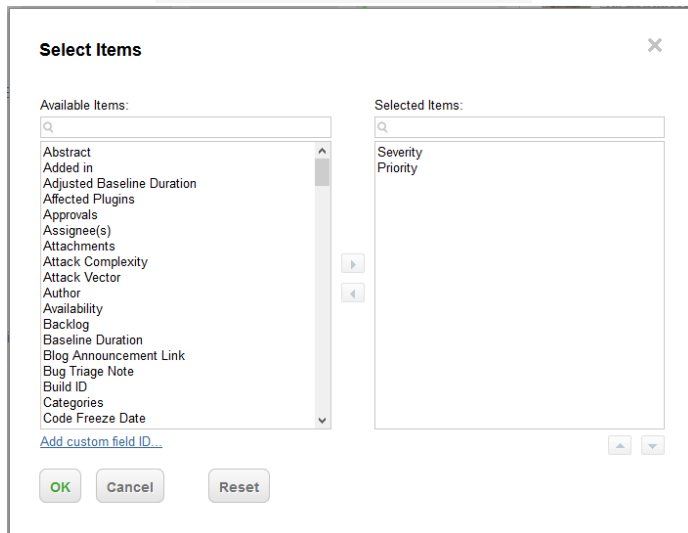
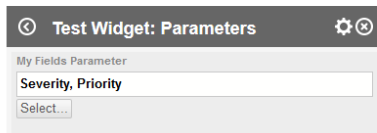
This example shows how to use a widget parameter of type Fields selector and how to define which fields are selected by default.

This is the content of widget's `parameters.vm` file:

```
#set($myFields = $objectFactory.newSet())

#set($void = $myFields.add("priority"))
#set($void = $myFields.add("severity"))

$parameters.put("myFieldsParameter", $factory.fields("My Fields Parameter").fields($myFields).build())
```



6.5 Get Historical Object by Date

The following code example shows how to get a respective revision for a given date, followed by how to use that revision

to get historical data of an object.

Example displays **number of work items** linked to plans when **started and finished**.

```
<table class="polarion-rpw-table-content">
  <tbody>
    <tr class="polarion-rpw-table-header-row">
      <th>Plan #</th>
      <th># of Work Items at Start</th>
      <th># of WIs at End</th>
    </tr>

    #foreach($plan in $transaction.plans().search())
      #set($revisionStarted =
$transaction.revisions().getBy().dateOrAfter($plan.fields().startedOn().get()).fields().id().get())
      #set($planAtStart =
$transaction.plans().getBy().revision($revisionStarted).path($plan.getReference().toPath()))
      #set($revisionFinished =
$transaction.revisions().getBy().dateOrAfter($plan.fields().finishedOn().get()).fields().id().get())
      #set($planAtEnd =
$transaction.plans().getBy().revision($revisionFinished).path($plan.getReference().toPath()))

      <tr class="polarion-rpw-table-content-row">
        <td>$plan.fields().name().get()</td>
        <td>$planAtStart.items().size()</td>
        <td>$planAtEnd.items().size()</td>
      </tr>

    #end

  </tbody>
</table>
```

Getting historical data for a revision:

```
<table class="polarion-rpw-table-content">
  <tbody>
    <tr class="polarion-rpw-table-header-row">
      <th>Plan #</th>
      <th># of Work Items at Start</th>
      <th># of WIs at End</th>
    </tr>

    #set($counter=1)

    #foreach($plan in $transaction.plans().search().limit(19))
      #if($plan.fields().startedOn().get() && $plan.fields().finishedOn().get() &&
$plan.fields().name().get().contains("Team"))
        #set($revisionStarted =
$transaction.revisions().getBy().dateOrAfter($plan.fields().startedOn().get()).fields().id().get())
        #set($planAtStart =
$transaction.plans().getBy().revision($revisionStarted).path($plan.getReference().toPath()))
        #set($revisionFinished =
$transaction.revisions().getBy().dateOrAfter($plan.fields().finishedOn().get()).fields().id().get())
        #set($planAtEnd =
$transaction.plans().getBy().revision($revisionFinished).path($plan.getReference().toPath()))
        #if(!$planAtStart.isUnresolvable() && !$planAtEnd.isUnresolvable())

          <tr class="polarion-rpw-table-content-row">
            <td>$counter</td>
            <td>$planAtStart.items().size()</td>
            <td>$planAtEnd.items().size()</td>
          </tr>

          #set($counter=$math.add($counter, 1))
        #end
      #end
    #end

  </tbody>
</table>
```

6.6 How to Use Page Parameters in Velocity and Queries

6.6.1 String Parameters

Examples below refer to a page parameter with ID "Name" of type "String" and value "John Doe".

To access the instance:

```
$pageParameters.Name => StringParameterImpl: Name, John Doe
```

To obtain the value:

```
$pageParameters.Name.value => John Doe
```

To obtain the value usable in SQL queries:

```
$pageParameters.Name.toSql => John Doe
```


To obtain the value usable in Lucene queries:

```
$pageParameters.Name.toLucene => "John Doe"
```

6.6.2 Integer Parameters

Examples below refer to a page parameter with ID "Year" of type "Integer" and value "2014".

To access the instance:

```
$pageParameters.Year => IntegerParameterImpl: Year, 2014
```

To obtain the value:

```
$pageParameters.Year.value => 2014
```

To obtain the value usable in SQL queries:

```
$pageParameters.Year.toSql => 2014
```

To obtain the value usable in Lucene queries:

```
$pageParameters.Year.toLucene => 2014
```

6.6.3 Boolean Parameters

Examples below refer to a page parameter with ID "Advanced" of type "Boolean" and value "Yes".

To access the instance:

```
$pageParameters.Advanced => BooleanParameterImpl: Advanced, true
```

To obtain the value:

```
$pageParameters.Advanced.value => true
```

To obtain the value usable in SQL queries:

```
$pageParameters.Advanced.toSql => true
```

To obtain the value usable in Lucene queries:

```
$pageParameters.Advanced.toLucene => true
```

6.6.4 Enumeration Parameters

Examples below refer to a page parameter with ID "reqtype" of type "Enumeration" and values "Functional" and "Business".

To access the instance:

```
$pageParameters.reqtype => EnumParameterImpl: Requirement Types, [functional, business]
```

To obtain the list of values:

```
$pageParameters.reqtype.values => [ProxyEnumOption id:functional, ProxyEnumOption id:business]
```

Returned list is usable in #foreach loop. List is returned also if the enumeration is not multi-valued.

To obtain the first value (or the sole value for single-value enumeration):

```
$pageParameters.reqtype.singleValue => ProxyEnumOption id:functional
```

To obtain the value usable in SQL queries:

```
$pageParameters.reqtype.toSql => ('functional', 'business')
```

Usage example: WHERE C_TYPE IN \$pageParameters.reqtype.toSql

To obtain the value usable in Lucene queries:

```
$pageParameters.reqtype.toLucene => (functional business)
```

Usage example: reqtype:\$pageParameters.reqtype.toLucene

6.6.5 Custom Enumeration Parameters

Examples below refer to a page parameter with ID "color" of type "Custom Enumeration" names "Red", "Green", "Blue" and values "#f00", "#0f0".

To access the instance:

```
$pageParameters.color => com.polarion.alm.shared.api.model.rp.parameter.impl.enumeration.CustomEnumParameterImpl@430f315
```

To obtain the list of values:

```
$pageParameters.color.values => [#f00, #0f0, #00f]
```

Returned list is usable in #foreach loop. List is returned also if the enumeration is not multi-valued.

To obtain the first value (or the sole value for single-value enumeration):

```
$pageParameters.color.singleValue => #f00
```

To obtain the list of names:

```
$pageParameters.color.names => [Red, Green, Blue]
```

Returned list is usable in #foreach loop. List is returned also if the enumeration is not multi-valued.

To obtain the first name (or the sole name for single-value enumeration):

```
$pageParameters.color.singleName => Red
```

6.6.6 Date Parameters

Examples below refer to a page parameter with ID "From" of type "Date" and absolute date March 20th 2015.

To access the instance:

```
$pageParameters.From => DateParameterImpl: From, Fri Mar 20 00:00:00 CET 2015
```

To obtain the value:

```
$pageParameters.From.value => Fri Mar 20 00:00:00 CET 2015
```

To obtain the value usable in SQL queries:

```
$pageParameters.From.toSql => 2015-03-20
```

To obtain the value usable in Lucene queries:

```
$pageParameters.From.toLucene => 20150320
```

6.6.7 Usage in "Lucene + Velocity" and "SQL + Velocity" Queries

Method calls `toSql()` and `toLucene()` are automatically added when page parameters are used in "Lucene + Velocity" and "SQL + Velocity" queries. So given a page parameter with ID "Year", one can use the query for all items resolved in that year:

```
resolvedOn: [{pageParameters.Year}0101 TO {pageParameters.Year}1231]
```

For more information see Javadoc for methods defined by interface

```
com.polarion.alm.shared.api.model.rp.parameter.ParameterUsableInQuery.
```

7 Page Script

The Page Script is executed before the widgets and it can be used to:

- initialize some variables which can then be used by the widgets
- add page parameters to which the widget parameters can be bound.

The Page Script can use the same Velocity context variables as widgets with following exceptions:

- **parameters** is not available
- **pageContext** is available; it is the holder for the context data
- **factory** is available for creating the scripted page parameters
- **scriptedPageParameters** is available for adding the scripted page parameters

The results of executing the Page Script can be displayed using the "Show Results" action in the sidebar toolbar (can be useful for debugging)

The errors and warning logged by Velocity can be displayed using the "Show Problems" action in the sidebar menu, which is displayed only if there are some problems reported.

7.1 Page Context

Page Context is the holder of the variables prepared by the Page Script and then accessed by the Scripted widgets. It can be modified only by the Page Script.

The rendering of the widgets is running concurrently in some cases, so the variables placed in the Page Context must be thread-safe objects if accessed by more than 1 widget in the page.

The API objects returned from transaction like `WorkItem` or `Document` should not be stored to the Page Context, because they are usable only inside the given transaction, and the rendering of the widgets is executed in different transactions in some cases. Instead, store references like `WorkItemReference`, which you get from the `WorkItem.getReference()` method and later in the widget. Use the `$reference.get($transaction)` to get the `WorkItem` again.

7.1.1 Page Context Example

This example Page Script shows how to prepare common parts of SQL queries which can be then be used by multiple widgets to display the same data in different ways:

```
#set($uncoveredReqWhere = $objectFactory.newStringBuilder)
$!uncoveredReqWhere.append("and C_TYPE in $pageParameters.reqtypes.toSql ${esc.n}").null

#if(!$pageParameters.version.values.empty)
    $!uncoveredReqWhere.append("and CUSTOM_FIELD.FK_WORKITEM = WORKITEM.C_PK ${esc.n}").null
    $!uncoveredReqWhere.append("and CUSTOM_FIELD.C_NAME = 'targetVersion' ${esc.n}").null
    $!uncoveredReqWhere.append("and CUSTOM_FIELD.C_STRING_VALUE = '$pageParameters.version.singleValue().id()' ${esc.n}").null
#end

#if(!$pageParameters.status.values.empty)
    $!uncoveredReqWhere.append("and C_STATUS IN $pageParameters.status.toSql ${esc.n}").null
#end

#set($coveredReqWhere = $objectFactory.newStringBuilder)
$!coveredReqWhere.append($uncoveredReqWhere).null

#set($testQuery = $objectFactory.newStringBuilder)
$!testQuery.append("select TEST.C_PK from WORKITEM TEST, STRUCT_WORKITEM_LINKEDWORKITEMS LINK ${esc.n}").null
$!testQuery.append("    where LINK.FK_WORKITEM = WORKITEM.C_PK ${esc.n}").null
$!testQuery.append("    and LINK.FK_P_WORKITEM = TEST.C_PK ${esc.n}").null
$!testQuery.append("    and LINK.C_ROLE = '$pageParameters.verifiesLinkRole.toSql()'").null

$!uncoveredReqWhere.append("and not exists ($!testQuery)").null
```

```

$!coveredReqWhere.append("and exists ($!testQuery)").null

$pageContext.put("uncoveredReqWhere", "$uncoveredReqWhere")
$pageContext.put("coveredReqWhere", "$coveredReqWhere")

#if (!$pageParameters.version.values.empty)
    $!pageContext.put("joinCustomField", ",CF_WORKITEM_CUSTOM_FIELD").null
#end

## show the values in the result for debugging
<pre>
uncoveredReqWhere:

$uncoveredReqWhere

coveredReqWhere:

$coveredReqWhere
</pre>

```

The Example was developed for the Drive Pilot demo project and uses the following page parameters:

- reotypes - Requirement Item Types - multi-enumeration parameter for the *workitem-type* enumeration
- version - Version - enumeration parameter for Plans with query: *template.id:release*
- status - Status - enumeration parameter for the *status* enumeration

The results of this page script:

```

uncoveredReqWhere:

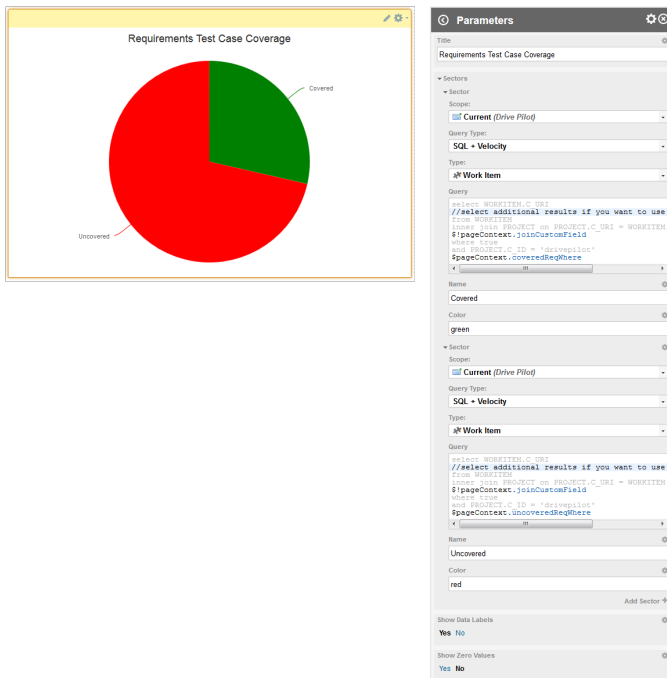
and C_TYPE in ('systemRequirement', 'softwareRequirement', 'mechanicalRequirement', 'electricalRequirement')
and CUSTOM_FIELD.FK_WORKITEM = WORKITEM.C_PK
and CUSTOM_FIELD.C_NAME = 'targetVersion'
and CUSTOM_FIELD.C_STRING_VALUE = 'Version_2_0'
and C_STATUS IN ('approved')
and not exists (select TEST.C_PK from WORKITEM TEST, STRUCT_WORKITEM_LINKEDWORKITEMS LINK
    where LINK.FK_WORKITEM = WORKITEM.C_PK
    and LINK.FK_P_WORKITEM = TEST.C_PK
    and LINK.C_ROLE = 'verifies')

coveredReqWhere:

and C_TYPE in ('systemRequirement', 'softwareRequirement', 'mechanicalRequirement', 'electricalRequirement')
and CUSTOM_FIELD.FK_WORKITEM = WORKITEM.C_PK
and CUSTOM_FIELD.C_NAME = 'targetVersion'
and CUSTOM_FIELD.C_STRING_VALUE = 'Version_2_0'
and C_STATUS IN ('approved')
and exists (select TEST.C_PK from WORKITEM TEST, STRUCT_WORKITEM_LINKEDWORKITEMS LINK
    where LINK.FK_WORKITEM = WORKITEM.C_PK
    and LINK.FK_P_WORKITEM = TEST.C_PK
    and LINK.C_ROLE = 'verifies')

```

Example widget using the variables. The parameters of the widget are shown on the right side:



7.2 Scripted Page Parameters

Page Script can add additional page parameters on top of the page parameters defined in the "Page Parameters" sidebar. Page Script cannot add a page parameter with same ID as a parameter already defined in the sidebar and an error will be logged in the page script error log if it tries to.

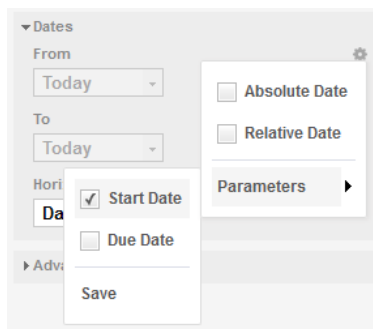
7.2.1 Scripted Page Parameters example

This example Page Script shows how to add 2 date type page parameters with values taken from Plan Start Date and Due Date fields. The example is meant to be used in a Plan report.

```
#set($startDate = $plan.fields.startDate.getIfCan)
#set($startDateBuilder = $factory.date("Start Date"))
#if ($!startDate)
    $!startDateBuilder.absolute($!startDate)
#end
$!scriptedPageParameters.put("startDate", $startDateBuilder.build())

#set($dueDate = $plan.fields.dueDate.getIfCan)
#set($dueDateBuilder = $factory.date("Due Date"))
#if ($!dueDate)
    $!dueDateBuilder.absolute($!startDate)
#end
$!scriptedPageParameters.put("dueDate", $dueDateBuilder.build())
```

These parameters can be then used for any date parameters in widgets, for example in chart widgets as shown below:



8 Dependencies between widget parameters

It is possible to define dependencies between widget parameters. When a report designer changes the value of a widget parameter that is marked as 'dependency source', any other parameters that are marked as 'dependency target' can be reconfigured accordingly.

8.1 Example

A widget has two parameters:

- the first is the object selector parameter where user can select a Work Item,
- the second is an enumeration parameter for selecting link roles.

The second parameter should provide possibility to select link role from project when defined the Work Item.

Complete implementation of this example is available in Work Records Widget examples (java and velocity).

8.1.1 Java widget

Parameters definition:

```
parameters.put(PARAMETER_USER_STORY, factory.objectSelector("User Story").allowedPrototypes(PrototypeEnum.WorkItem).dependencySource(true).build());
parameters.put(PARAMETER_LINK_ROLE, factory.enumeration("Link Role", "workitem-link-role").dependencyTarget(true).build());
```

For implementing a dependency method

`com.polarion.alm.shared.api.model.rp.widget.RichPageWidget.processParameterDependencies(RichPageWidgetDependenciesContext)` should be overridden in a widget class:

```
ObjectSelectorParameter parameter = context.parameter(WorkRecordReportWidget.PARAMETER_USER_STORY);
WorkItem workItem = (WorkItem) parameter.value();
Scope scope = (workItem == null) ? null : workItem.getReference().scope();
```

```
EnumParameter parameter = context.parameter(WorkRecordReportWidget.PARAMETER_LINK_ROLE);
parameter.set().scope(scope);
```

8.1.2 Velocity widget

Parameters definition

```
$parameters.put("userStory", $factory.objectSelector("User Story").allowedPrototypes(["WorkItem"]).dependencySource(true).build())
$parameters.put("linkRole", $factory.enumeration("Link Role", "workitem-link-role").dependencyTarget(true).build())
```

Implementation of dependency is defined in `dependencies.vm` file

```
#set($userStoryParameter = $parameters.userStory)
#set($linkRoleParameter = $parameters.linkRole)
```

```

#set($workItem = $userStoryParameter.value)

#if (!$workItem)
  #set($scope = $null)
#else
  #set($scope = $workItem.getReference().scope())
#end

$linkRoleParameter.set().scope($scope)

```

9 Actions in widgets

It is possible to create a button in widget that triggers a server side action on click: a button that starts plan, for example.

9.1 How it works

Rendered widget contains elements with defined special attribute (`RichPageWidget.ATTRIBUTE_ACTION_ID`).

When user clicks on it, a request is sent to the server and `RichPageWidget.executeAction(RichPageWidgetActionContext)` is triggered

Widget developer can handle the event by implementing `executeAction` in the widget.

Note: standard action handling is persisting some changes, so requesting of user confirmation up front is a good practice. (see

`RichPageWidget.ATTRIBUTE_CONFIRM_* keys`)

Also widget developer can define whether the page should be refreshed after the action executes, or if just the widget will be re-rendered, which can be done by calling `RichPageWidgetActionContext.refresh(boolean)`.

It's important to understand that there can be several widgets of the same kind on the same page, so identification of exactly what should be done should be based on the value of `ATTRIBUTE_ACTION_ID` and the widget parameters.

Action handling is available only for Java Rich Page widgets.

An updatable version of model objects can be received by `ModelObjectBase.getUpdatable(WriteTransaction)`

9.2 Example

```

private void renderAction(@NotNull HtmlContentBuilder htmlContentBuilder, @NotNull DurationValue
total) {
    HtmlTagBuilder a = htmlContentBuilder.tag().a();
    a.append().text("Update time spent in the user story");
    a.attributes().byName(RichPageWidget.ATTRIBUTE_ACTION_ID, total.toString());
    a.attributes().byName(RichPageWidget.ATTRIBUTE_CONFIRM_TITLE, "Update time spent");
    a.attributes().byName(RichPageWidget.ATTRIBUTE_CONFIRM_TEXT, "Do you want to update time
spent?");
}

public void executeAction(@NotNull RichPageWidgetActionContext context) {
    ObjectSelectorParameter parameter = context.parameter(PARAMETER_USER_STORY);
    WorkItem workItem = (WorkItem) parameter.value();
    if (workItem == null) {
        return;
    }
    Duration value;
    try {
        value = new Duration(context.actionId());
    } catch (Exception e) {
        return;
    }
    UpdatableWorkItem updatableWorkItem = workItem.getUpdatable(context.transaction());
    updatableWorkItem.fields().timeSpent().set(value);
    updatableWorkItem.save();
    context.refresh(true);
}

```

Complete implementation of this example is available in the Java Work Records Widget example.

10 Velocity Context Reference

10.1 Available Everywhere

10.1.1 Polaron Objects

- **transaction**
 - `com.polarion.alm.shared.api.transaction.ReadOnlyTransaction`
- **localization**
 - `com.polarion.alm.shared.api.utils.SharedLocalization`
- **me**
 - `java.lang.String`
 - current user ID (shortcut for `$transaction.context.currentUserId`)
- **objectFactory**

- com.polarion.alm.shared.api.utils.velocity.ObjectFactory

10.1.2 Polarion Services (from Polarion Open Java API)

- **trackerService**
 - com.polarion.alm.tracker.ITrackerService
- **projectService**
 - com.polarion.alm.projects.IProjectService
- **securityService**
 - com.polarion.platform.security.ISecurityService
- **platformService**
 - com.polarion.platform.IPlatformService
- **testManagementService**
 - com.polarion.alm.tracker.ITestManagementService

10.1.3 Velocity Generic Tools

- **date**
 - <http://velocity.apache.org/tools/releases/2.0/javadoc/org/apache/velocity/tools/generic/DateTool.html>
- **math**
 - <http://velocity.apache.org/tools/releases/2.0/javadoc/org/apache/velocity/tools/generic/MathTool.html>
- **number**
 - <http://velocity.apache.org/tools/releases/2.0/javadoc/org/apache/velocity/tools/generic/NumberTool.html>
- **esc**
 - <http://velocity.apache.org/tools/releases/2.0/javadoc/org/apache/velocity/tools/generic/EscapeTool.html>
- **alternator**
 - <http://velocity.apache.org/tools/releases/2.0/javadoc/org/apache/velocity/tools/generic/AlternatorTool.html>
- **lists**
 - <https://velocity.apache.org/tools/2.0/apidocs/org/apache/velocity/tools/generic/ListTool.html>
- **sorter**
 - <http://velocity.apache.org/tools/releases/2.0/javadoc/org/apache/velocity/tools/generic/SortTool.html>
- **mill**
 - <http://velocity.apache.org/tools/releases/2.0/javadoc/org/apache/velocity/tools/generic/IteratorTool.html>

10.2 Available in Velocity Widgets and Page Script

All of these are available also in "Lucene + Velocity" and "SQL + Velocity" queries.

10.2.1 Velocity Widget Objects

- **renderingContext**
 - com.polarion.alm.shared.api.model.rp.widget.RichPageRenderingContext in Page Script
 - com.polarion.alm.shared.api.model.rp.widget.RichPageWidgetRenderingContext in other cases
- **widgetContext**
 - synonym for renderingContext
 - not available in Page Script
- **pageContext**
 - java.util.Map<Object, Object>
 - not available in details.vm and parameters.vm
- **parameters**
 - com.polarion.alm.shared.api.utils.collections.ReadOnlyStrictMap<String, ? extends com.polarion.alm.shared.api.model.rp.parameter.RichPageParameter>
 - not available in details.vm and the Page Script
- **pageParameters**
 - com.polarion.alm.shared.api.utils.collections.ReadOnlyStrictMap<String, ? extends com.polarion.alm.shared.api.model.rp.parameter.RichPageParameter>
 - not available in details.vm and parameters.vm
- **scriptedPageParameters**
 - com.polarion.alm.shared.api.utils.collections.StrictMap<String, ? extends com.polarion.alm.shared.api.model.rp.parameter.RichPageParameter>
 - available only in Page Script
- **urlParameters**
 - com.polarion.alm.shared.api.utils.collections.ReadOnlyStrictMap<String, String>
 - not available in details.vm and parameters.vm
- **page**
 - current page: com.polarion.alm.shared.api.model.rp.RichPage
 - available for stand alone pages, not available in details.vm and parameters.vm
- **plan**
 - current Plan object: com.polarion.alm.shared.api.model.plan.Plan
 - available only on Plans that uses Rich Page technology

- **object**
 - current object: com.polarion.alm.shared.api.model.ModelObject
 - reference to plan or page (dependent on context)
- **factory**
 - com.polarion.alm.shared.api.model.rp.parameter.ParameterFactory
 - available in parameters.vm and the Page Script

11 Examples of using ParameterFactory in Velocity

11.1 Adding scripted page parameters in Page Script

```

$!scriptedPageParameters.put("myString", $factory.string("My String").value("my value").build())
$!scriptedPageParameters.put("myInteger", $factory.integer("My Integer").value(3).build())
$!scriptedPageParameters.put("myBoolean", $factory.bool("My Boolean").value(true).build())
$!scriptedPageParameters.put("mySingleSeverity", $factory.enumeration("My Single Severity", "severity").singleValue("major").build())
$!scriptedPageParameters.put("myMultiSeverity", $factory.enumeration("My Multi Severity",
"severity").allowMultipleValues(true).values(["major", "critical"]).build())
$!scriptedPageParameters.put("myRelativeDate", $factory.date("My Relative Date").relative(-12).build())
$!scriptedPageParameters.put("myAbsoluteDate", $factory.date("My Absolute Date").absolute($date.toDate("yyyy-MM-dd",
"2015-01-02")).build())
$!scriptedPageParameters.put("myPlan", $factory.objectSelector("My
Plan").allowedPrototypes(["Plan"]).objectPath("drivepilot/Version_1_0").build())

```

The other parameter types cannot be used as page parameters.

11.2 Adding widget parameters in parameters.vm

All the above mentioned examples can be used. Just replace *\$!scriptedPageParameters* with *\$!parameters*

Examples of some other parameter types:

```

$!parameters.put("myScript", $factory.script("My Script").value("${esc.d}me").build())
$!parameters.put("myTimeAxis", $factory.timeAxis("My Time Axis").fromRelative(-28).toRelative(0).scaleWeek().build())
$!parameters.put("myFields", $factory.fields("My Fields").fields($mySetOfFields).build())

```

Example of combining *DataSetParameter* with *FieldsParameter* and *SortingParameter* in the way they are used in the Table widget:

```

#set($myColumns = $factory.fields("My Columns").build())
#set($mySorting = $factory.sorting("My Sorting").build())
$!parameters.put("myDataSet", $factory.dataSet("My Data Set").add("myColumns", $myColumns).add("mySorting", $mySorting).build())

```

Example of Custom enumeration parameter:

```

$parameters.put("teams", $factory.customEnum("Teams").addEnumItem("mainDev", "Main Development Team").addEnumItem("mainQa",
"Main QA Team").addEnumItem("extDev", "External Dev Team").addEnumItem("extQa", "External QA
Team").allowMultipleValues(true).values(["mainDev", "mainQa"]).build())

```

12 Predefined Velocity Macros

There is set of predefined Velocity Macros that can be used in Script widgets.

Plain Text box

```
#message("some plain text")
```

some plain text

Information Message box

```
#info("something")
```

Info: something

Warning Message box

```
#warning("something")
```

Warning: something

Error Message box

```
#error("something")
```

Error: something

Load CSS

```
#loadCss("http://mywebsite.com/myCSS.css")
```

Load CSS from Widget Resources

```
#loadWidgetCss("resources/myCSS.css")
```

Load Java Script File

```
#loadJs("http://mywebsite.com/myJS.js")
```

Load Java Script File from Widget Resources

```
#loadWidgetJs("resources/myJS.js")
```

13 Enabling the Work Item Properties Sidebar in a Custom Widget

Enabling the Work Item Properties Sidebar is done in two parts:

1. Set up the initial java script configuration, to associate part or all of the widget with the Properties Sidebar. This should be done at the beginning of rendering and is accomplished by using `com.polarion.alm.shared.api.model.rp.widget.PropertiesSidebarConfiguration`
2. Set up the configuration to have the Properties Sidebar open on click during the rendering of each individual item. This is done by adding html attributes to the elements where the item is rendered by using `com.polarion.alm.shared.api.model.rp.widget.PropertiesSidebarConfiguredContainer`

An example is provided in the Polarion SDK examples under `com.polarion.example.widget`.

13.1 What HTML attributes are configured on a Work Item so that the Work Items Sidebar is displayed

- Path to the item (for Work Items: project id/work item id):
`com.polarion.alm.shared.api.model.rp.widget.PropertiesSidebarConfiguredContainer.Attributes.referencePath()`
- Css class for styling before clicking:
`com.polarion.alm.shared.api.model.rp.widget.PropertiesSidebarConfiguredContainer.Attributes.clickableCssClass()`
- Css class for highlighting the item once the sidebar is open:
`com.polarion.alm.shared.api.model.rp.widget.PropertiesSidebarConfiguredContainer.Attributes.highlightingCssClass()`
- (Optional) Attribute for associating the list of fields displayed in the sidebar for a group of Work Items:
`com.polarion.alm.shared.api.model.rp.widget.PropertiesSidebarConfiguredContainer.Attributes.fieldsConfigurationKey()`

Also see subchapter ["13.2 Showing Different Fields for Different Groups of Work Items"](#).

They can be accessed from `com.polarion.alm.shared.api.model.rp.widget.PropertiesSidebarConfiguredContainer.attributes()`;

13.2 Showing Different Fields for Different Groups of Work Items

This can be done by providing an initial configuration using

```
com.polarion.alm.shared.api.model.rp.widget.PropertiesSidebarConfiguration.fieldsConfiguration(Map<String, List<String>>).
```

Example:

For the Work Items Board Widget, which has a structure with Swimlanes containing User Stories and Cards containing Implementation tasks or Issues, we would like to show a different set of fields in the sidebar:

- Status, severity and priority for the swimlane elements.
- Status and author for the card elements.

In this case, during the initial configuration, we would have:

```
...
Map<String, List<String>> configurationMap = new HashMap<>();
List<String> swimlaneFields = Arrays.asList(new String[] { "status", "severity", "priority" });
configurationMap.put("swimlane", swimlaneFields);
List<String> cardFields = Arrays.asList(new String[] { "status", "author" });
configurationMap.put("card", cardFields);
context.propertiesSidebarConfiguration().fieldsConfiguration(configurationMap).addTo(builder);
..
```

And later during the rendering of cards:

```
...
configuredContainer.openSidebarOnClick(tr).reference(wiReference).fieldsConfigurationKey("card"); << When clicking on this item, the
sidebar will display "status", "author".
...
Or during the rendering of swimlanes:
```

```
...
configuredContainer.openSidebarOnClick(tr).reference(wiReference).fieldsConfigurationKey("swimlane"); << When clicking on this
item, the sidebar will display "status", "severity", "priority".
```

Note: The different field lists could be read from different widget field parameters.

14 Customizing the Kanban Board

The Kanban board is made up of Work Item cards grouped by status columns.

You can customize the appearance of the cards by adding custom velocity snippets. You can change the color of the card, remove the status border color, show different Work Item fields, re-arrange the fields already rendered and *more*. The custom Velocity scripts will be rendered instead of the default java scripts.

Default velocity snippets are included with the widget under the "Board Script" and "Card Script" parameters and are a great starting point.

(The sample scripts mimic the java rendered output out of the box.)

First, set the "Enable Customization" widget parameter to "Yes". The "Board Script" and "Card Script" parameters appear below.

The "Board script" is inserted at the beginning of the board and should always be used to add custom css styles and/or java script common for all your cards.

(This avoids having to duplicate them for each card.)

Warning: The styles inserted in this script are applied to the whole page.

The "Card Script" defines the content that appears within the cards. The "Card Script" Velocity script option allows for the following additional objects:

- **\$workItem** - The Work Item rendered in the card.
- **\$columnIndex** - The column that the card is rendered on.

Info: The Work Item Properties sidebar and the ability to drag and drop are still available when customizations are enabled.

You can always revert to using the internal Java rendering by setting "Enable Customization" to "No".

Here are a few examples on how to customize the look and content of your Kanban board's cards:

14.1 Examples

14.1.1 How to change the card color

1. Open a page containing a Kanban board in edit mode.
2. Open the "Kanban Board: Parameters" sidebar.
3. Click on "Advanced" at the bottom and click "Yes" under "Enable Customizations".
4. Click on "Board Script". (For a full screen view of the script hit **F11**. Hit **Esc** to exit full screen view.)
5. Change the background color of the card by adding "background-color: #d0d2e0;" to ".kanbanWorkItemDiv".
6. Change the gradient of a long title by changing the "background" property in ".kanbanWorkItemTitleGradientDiv" to "background: linear-gradient(to bottom, rgba(255, 255, 255, 0), #cffffff) repeat scroll 0 0 rgba(0, 0, 0, 0);"
7. Enable the **Work Item Properties Sidebar** by adding:

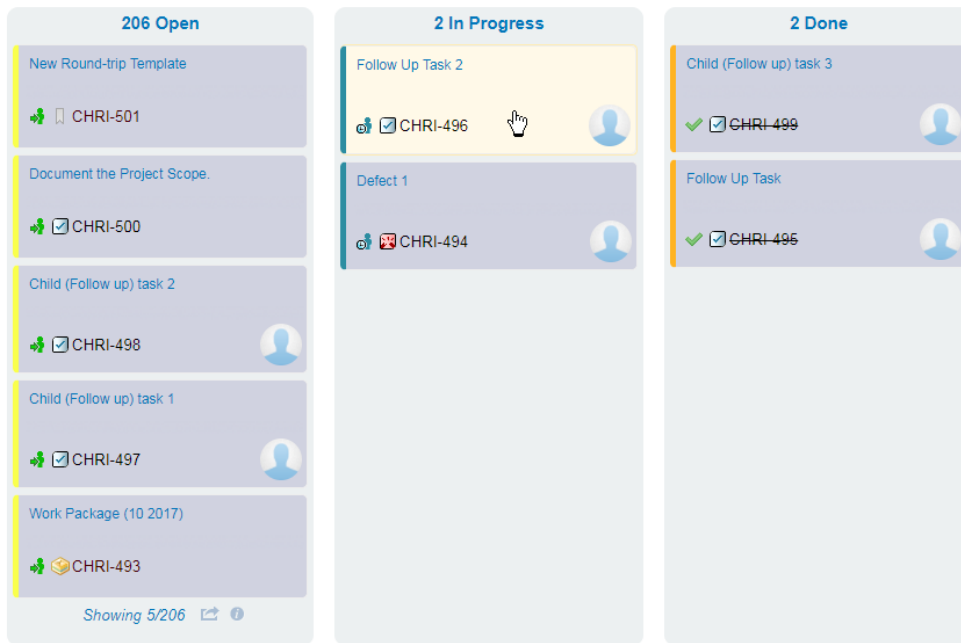
```
.polarion-sidebar-highlight .kanbanWorkItemDiv {
  background-color: #fff9e8;
}
```

The only changes to the default "Board Script" are the **bold** lines in the example below:

```
...
.kanbanWorkItemDiv {
  padding: 7px 3px 3px 7px;
  box-sizing: border-box;
  border-left: 5px solid;
  border-right: 1px solid #EEEEEE;
  border-top: 1px solid #EEEEEE;
  border-bottom: 1px solid #EEEEEE;
  border-radius: 5px;
  position: relative;
  background-color: #d0d2e0;
}
...
.kanbanWorkItemTitleGradientDiv {
  bottom: 0px;
  height: 12px;
  left: 0;
  position: absolute;
  width: 100%;
  display: inline-block;
  background: linear-gradient(to bottom, rgba(255, 255, 255, 0), #d0d2e0) repeat scroll 0 0 rgba(0, 0, 0, 0);
  filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#00FFFFFF', endColorstr='#FFFFFF', GradientType=0);
}
...
.polarion-sidebar-highlight .kanbanWorkItemDiv {
  background-color: #fff9e8;
}
...
```

8. Click "Apply".

The color of the cards will change to the following:



(No changes are required to the default "Card Script".)

14.1.2 How to add a Work Item's Tree Structure link

You can customize what content appears within the cards in "Card Script".

1. Open a page containing a Kanban board in edit mode.
2. Open the "Kanban Board: Parameters" sidebar.
3. Click on "Advanced" at the bottom and click "Yes" under "Enable Customizations".
4. Click on "Card Script". (For a full screen view of the script hit **F11**. Hit **Esc** to exit full screen view.)
5. Paste the **bold** table text below under the `<div class="kanbanWorkItemDiv" style="border-left-color: $wiStatusColor;">` tag.
6. Click "Apply".

```
...
<div class="kanbanWorkItemDiv" style="border-left-color: $wiStatusColor;">

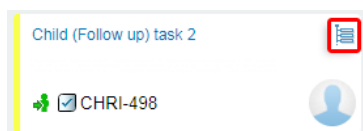
<table width="100%">
<tbody>
<tr>
<td class="kanbanWorkItemTitleDiv">
<div class="kanbanWorkItemTitleGradientDiv"></div>
<a href="$wiLink" target="_top">$workItem.fields().title().render()</a>
</td>
#set($treeLink =
$transaction.context().createPortalLink().project("$wiProjectId").workItems().query("id:$workItem.getReference().id()").toEncodedRelativeUrl()
#set($suffix = '&tree_depth=2&tab=tree')
<td style="vertical-align: top;">
<a href="$treeLink$suffix" target="_blank">

</a>
</td>
</tr>
</tbody>
</table>

<table class="kanbanWorkItemTable">
...

```

6. A tree icon with a link to the card's Work Item in **Tree** view is added to the card.



15 Frequently Asked Questions

Q: How do I get the ID of the current project?

A: In Velocity: `$page.reference.projectId`

Q: What to do when HTML table exported to PDF loses styles and/or there are no columns?

A: PDF export requires that HTML tags <tr>, <th> and <td> are closed with </tr>, </th> and </td>.