Report Tool Tips and Tricks

This document is a guide to help you become more productive with the Report Tool by showing you how to address some of the more common tasks that need to be done when designing a Report.

Query Tips

- Using the `select` attribute
  The `select` attribute enables you to filter down the set of properties for the Items returned from a query.

  ```
  <Item type="Part" action="get" select="part_number,part_description,part_co
  ```

- Using the `order_by` attribute
  The `order_by` attribute enables you to order the results of the query. The style sheet also supports sorting on the properties.

  ```
  <Item type="Part"
  action="get"
  select="part_number,part_description,part_cost"
  order_by="part_number"
  />
  ```

- Using the `condition` attribute
  The condition attribute enables you to specify a condition for the properties used as search criteria. All the conditions that are allowed in SQL are allowed as the values to the condition attribute.

  > **Important**
  >
  > The greater then and less then symbols are replaced with the two-letter pneumonic word. The < symbol is lt, and the > symbol is gt, >= is ge and so on.

```
<Item type="Part"
action="get"
select="part_number,part_description,part_cost"
order_by="part_number">
<part_number condition="like">%123%</part_number>
</Item>
```

- OR'ing property values
  The `<or>` tag provides a way to logically OR property values for the search criteria.

```
<Item type="Part"
action="get"
select="part_number,part_description,part_cost"
order_by="part_number">
<or>
<state>Released</state>
<state>In Review</state>
</or>
</Item>
```

- Return Relationships in the results
  You can include the `<Relationships>` tag and the relationship items you want to return.

```
<Item type="{@type}" id="{@id}" action="get" select="part_no,part_desc,part
<Relationships>
<Item type="Part EBOM" action="get" select="position,qty,related_id">
<related_id>
<Item type="Part" action="get" select="id,part_no,part_desc,part_cost">
<Relationships>
<Item type="Part AVL" action="get" select="avl_status,related_id(descriptio
</Relationships>
</Item>
</related_id>
</Item>
</Relationships>
</Item>
```

- Using Relationships as search criteria
  You can include the `<Relationships>` tag to describe the search criteria for the relationship items.

```
<Item type="{@type}" id="{@id}" action="get" select="part_no,part_desc,part
<Relationships>
<Item type="Part EBOM" action="get" select="position,qty,related_id">
<qry condition="gt">1000</qty>
</Item>
</Relationships>
</Item>
```

Stylesheet Tips

- Displaying a non-breaking white space
  To display a non-breaking white space, use the `<xsl:text>` tag.

```
<xsl:text disable-output-escaping="yes">&amp;nbsp;</xsl:text>
Or this:  
```

- Displaying the formatted text property
  To display formatted text (the marked-up text) property value in XSLT, use the disable-output-escaping attribute with the `<xsl:value-of>` tag. The marked-up text for the formatted text property is a complete HTML document and only the content for the BODY tag is required.

```
<xsl:value-of disable-output-escaping="yes" select=".//BODY/*"/>
```

- Displaying nested tables for relationships
  To display the relationship Items for nested tables, reduce the XPath to the relative path based on the context node for the `<xsl:for-each>` loop that the nested relationships table is in.

```
<xsl:for-each select="Relationships/Item">
<xsl:for-each select="related_id/Item/Relationships/Item">
</xsl:for-each>
</xsl:for-each>
```

- Resizing the **Report** window
  To resize the report window, add an onload callback function for the HTML page. Add the following code before the `<body>` tag:

```
<script>
onload = function()
{
top.window.resizeTo(800,600);
}
</script>
```

- Showing a different value based on the value for a property
  To show a different value in the Report based on the actual value of the property. For example, the actual value of the property is the hex number for a color, but you want to show the name of the color in text.

```
To show a different value in the Report based on the actual value of the pr
<xsl:choose>
<xsl:when test="mgr_opinion='#FF0000'">Red</xsl:when>
<xsl:when test="mgr_opinion='#FFFF00'">Yellow</xsl:when>
<xsl:when test="mgr_opinion='#00FF00'">Green</xsl:when>
</xsl:choose>
```

- Supporting European languages
  To show the results in a European language:

1. Set the encoding attribute for the root xml tag:

```
<?xml version="1.0" encoding="iso-8859-1" ?>
```

2. Set the encoding attribute for the output tag:

```
<xsl:output encoding="iso-8859-1" method="html"
omit-xml-declaration="yes" standalone="yes" indent="yes" />
```