

Basic Velocity Training

Part 2

Overview of Agenda

WHAT WILL BE COVERED?



Homework Discussion & Review



Homework Prompt ~ Title Page

EMBEDDED IMAGE & DYNAMIC TITLE PAGE CONTENT

Tasks:

1. Apply the report title using the global constant project object.
2. Apply a dynamic subtitle using a String report parameter.
3. Apply the full name of the current report runner using the \$userSource API interface.
4. Apply the embedded logo image and ensure renders when you run the report.

Homework Solution ~ Title Page

✓ *Embedded Image*



```
## Embedded Logo Image
<p class=MsoNormal align=center style='text-align:center'><!--[if gte vml
1]><v:shape style='width:142pt;height:44pt;visibility:visible;mso-wrap-style:square'>
<v:imagedata src="cid:image001.png" o:title=""/>
</v:shape><![endif]--></p>
```

✓ *Report Title*



```
## Title - Project Name
<p class=MsoNormal align=center style='text-align:center'><b><span style='font-size:
30.0pt;line-height:115%;font-family:"Arial",sans-serif'>$project.name</b></p>
```

✓ *Report Subtitle*



```
## Subtitle - String Parameter
<p class=MsoNormal align=center style='text-align:center'><span
style='font-size:16.0pt;line-height:115%;font-family:"Arial",sans-serif'>$!
reportSubtitle</p>
```

✓ *Report Runner*



```
## Author - Report Runner Name
<p class=MsoNormal align=center style='text-align:center'><span
style='font-size:10.0pt;line-height:115%;font-family:"Arial",sans-serif'>Author: $
userSource.currentUser.fullName</span></p>
```

Image base64 Embedded Image Definition

```
-----=_NextPart_01DA8992.E7786AB0
Content-ID: image001.png
Content-Transfer-Encoding: base64
Content-Type: image/png
```

```
<o:File HRef="item0001.xml"/>
<o:File HRef="image001.png"/>
<o:File HRef="props002.xml"/>
```

Criterion for Report Subtitle String Parameter

String Parameter	▼	Display: Enter Subtitle	Name: reportSubtitle	⊖
Add Report Criterion				⊕

Homework Prompt ~ Report Content Pages

ITEM TYPE CONDITIONALS AND CONTENT

Tasks:

1. For Containers items of Component, Set, and Folder (CMP, SET, FLD) display:
 - The item Name using a `<h1>` tag.
2. For Text items (TXT) display:
 - The item `<documentKey>` in `'[]'` braces followed by the item Name using a `<h2>` tag.
3. Create a case for an Item Type of your choice and display:
 - The item Name in a `<h3>` tag followed by a custom field in the following format:
 - “Field Label”: `<custom field value>`
4. Create an Else case condition and display:
 - The item `<globalId>` in static `'{}'` curly braces and followed by the Name in bold & italic format.

Homework Solution ~ Report Content Pages

- ✓ Format “CMP”, “SET”, & “FLD” <name> as <h1>.
- ✓ Format “TXT” [<documentKey>] <name> as <h2>
- ✓ Format chosen Item Type <name> in <h3> & “Custom Field Label”: <custom field>.
- ✓ Else {global Id} & style only the <name> bold and italic.

```
#foreach( $vDoc in $documentList )
  #set( $doc = $vDoc.document )
  #set( $docTypeKey = $doc.documentType.documentTypeKey )

  ## Containers
  #if( $docTypeKey == "CMP" || $docTypeKey == "SET" || $docTypeKey == "FLD" )
    <h1>$doc.name</h1>

  ## Text Items
  #elseif( $docTypeKey == "TXT" )
    <h2>[$doc.documentKey] $doc.name</h2>

  ## Chosen Item Type System Requirements
  #elseif( $docTypeKey == "SR" )
    #set( $rationale = '' )
    #set( $rationale = $velocityReportUtil.getValueForField($doc, "rationale", $dateFormat))
    <h3>$doc.name</h3>
    <p class=MsoNormal>Rationale: $!rationale</p>

  ## Unknown Item Types
  #else
    <p class=MsoNormal>{$doc.globalId}<span style='font-weight:bold;font-style:italic;'>$doc.name</span></p>

  #end
#end
```

Homework Prompt ~ Item Type Content Macro

CREATE ITEM TYPE MACROS FOR ITEM TYPE CONTENT

Each macro will contain your original code definition per item type handling as defined in your `#if` conditionals found in your primary `$documentList` `#foreach` loop.

Tasks:

1. For your Container item types (CMP, SET, FLD), you will define a new item type content macro called *#formatContainer()*
2. For Text item types (TXT)), you will define a new item type content macro called *#formatText()*
3. For your `#else` case of unmapped item types, create a macro called *#formatItem()*
 - In addition, for this item create a dynamic bullet list showing both the Upstream and Downstream related items in the following format:
 - "[<related item documentKey>]: <related item name>"

Homework Solution ~ Item Type Content Macros

CONTAINER & TEXT ITEM TYPE MACROS

- ✓ *#formatContainer macro call with the \$doc supplied as argument*



```
## Containers
#if( $docTypeKey == "CMP" || $docTypeKey == "SET" || $docTypeKey == "FLD" )
    #formatContainer($doc)
## Text Items
#elseif( $docTypeKey == "TXT" )
    #formatText($doc)
```

- ✓ *#formatText macro call with the \$doc supplied as argument*



- ✓ *#formatContainer macro definition*



```
#macro( formatContainer $doc )
    <h1>$doc.name</h1>
#end
```

- ✓ *#formatText macro definition*



```
#macro( formatText $doc )
    <h2>[$doc.documentKey] $doc.name</h2>
#end
```

Homework Solution ~ #formatItem Macro

#ELSE CASE MACRO

- ✓ *#formatItem macro call with the \$doc supplied as argument*



```
## Unknown Item Types
#else
    #formatItem($doc)
#end
```

- ✓ *#formatItem macro definition and with downstream and upstream relationships gathering and list creation.*



```
#macro( formatItem $doc )
  <p class=MsoNormal>{$doc.globalId} <span style="
    font-weight:bold;font-style:italic;">$doc.name</span>
  </p>

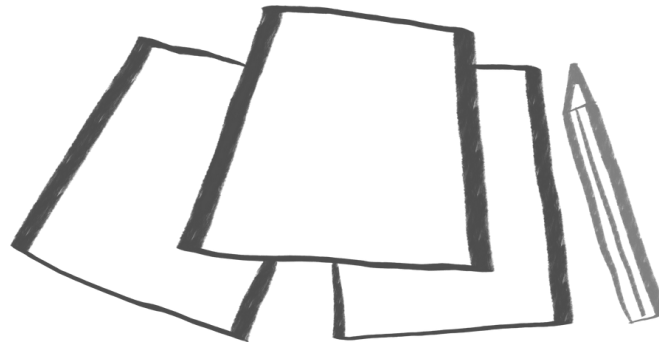
  <p class=MsoNormal><b>Downstream Items</b></p>
  #getRelationships($doc, true)
  #set($dnRels = $returnRels )
  <ul>
    #foreach($dnRel in $dnRels )
      #set($dnDoc = $dnRel.toDocument)
      <li>[$dnDoc.documentKey] $dnDoc.name</li>
    #end
  </ul>

  <p class=MsoNormal><b>Upstream Items</b></p>
  #getRelationships($doc, false)
  #set($upRels = $returnRels )
  <ul>
    #foreach($upRel in $upRels )
      #set($upDoc = $upRel.fromDocument)
      <li>[$upDoc.documentKey] $upDoc.name</li>
    #end
  </ul>

  <p class=MsoNormal>&nbsp;</p>
#end
```

Training Template Files Review

HOW TO UTILIZE VELOCITY TRAINING TEMPLATES



Word Template Structure

WORD (.doc) FORMAT REPORTS

- The "Jama_Velocity_Training_Word_Template.vm" is organized into defined sections as outlined in the image to the right to allow both ease of use and modification.
- The template provides you with a strong foundation of global variables, flags, and handling macros to streamline custom report development as much as possible.
- Each primary section includes an example implementation for you to either modify or utilize as a reference for building your own definition.

Developer Report Documentation & Parameters
Template Outline & Documentation
Document Initialization (export context, export container, etc...)
Tables of ... Initialization
Word Formatting Variables
<pre><html> <head> <style> Word Style Classes TOC List Defintions </style> </head></pre>
Template Macros <i>Rich Text Handling Macros</i> <i>Context Macros</i> <i>Item Link Macros</i> <i>Date/Time Macros</i> <i>Header / Footer Macro</i> <i>Custom Macros (for you to populate)</i> <i>Item Heading Macros (default item handling macro example)</i>
Report Body <pre><body> Title Page Table of... Output Document List Item Type Map </body> </html></pre>
Word Embedded Files Defintions

Excel Template Structure

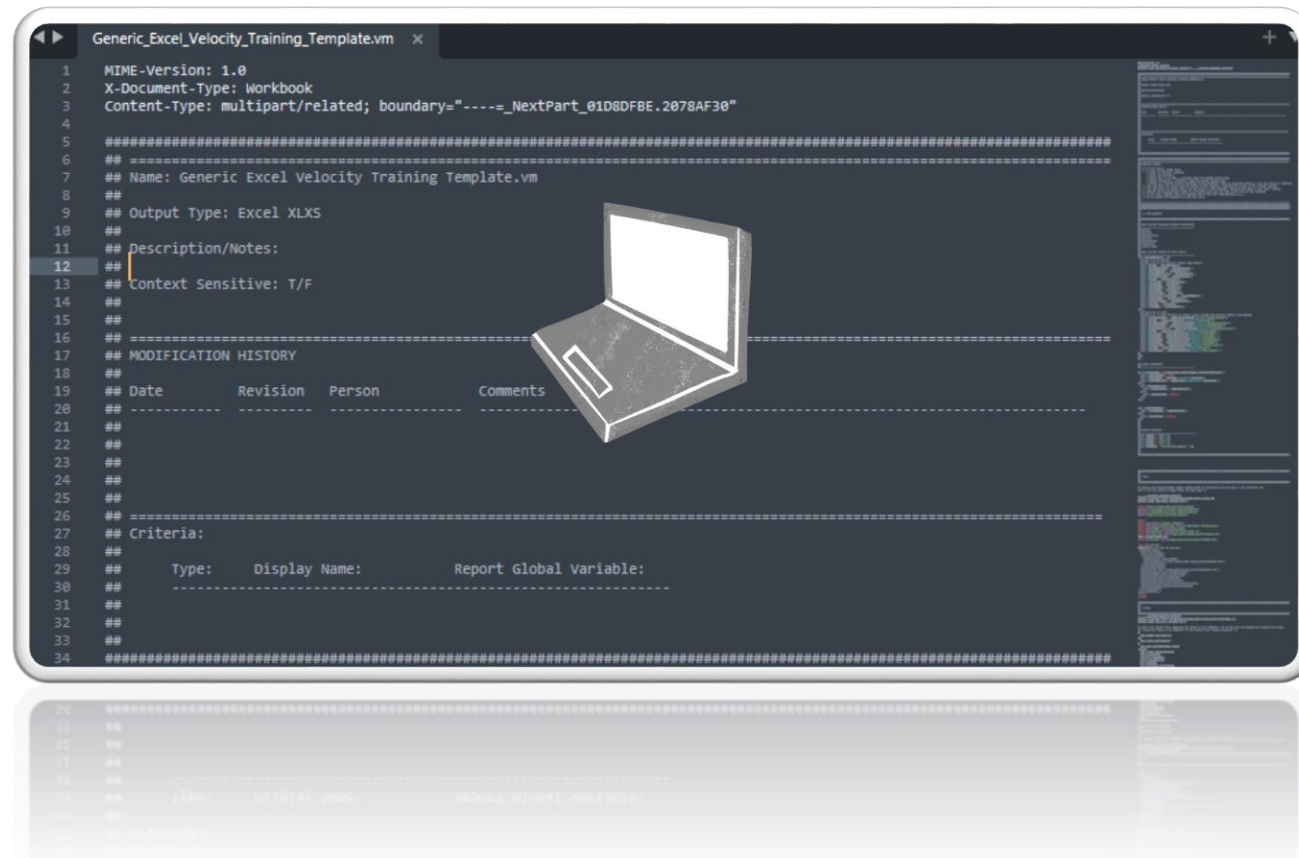
EXCEL (.xlsx) FORMAT REPORTS

- The "Jama_Velocity_Training_Excel_Template.vm" is for creating a single sheet Excel report of which you will find an outline of the defined sections in the image to the right.
- It recommend that your follow the "Template Steps" section found at the top of the template.

Developer Report Documentation & Parameters
Template Outline & Documentation
Data Sources, Parameters, General Constants
<pre><html> <head> Excel Workbook / Worksheets Defintions </head> </html></pre>
Excel Styles <i>Excel Style Classes</i>
Template Macros <i>Rich Text Handling Macro</i> <i>Cell Coloring Macros</i> <i>Item Handling Macros</i> <i>Date/Time Macros</i> <i>Custom Macros (for you to populate)</i>
<pre><html> <head> </head> Report Body <body> Sheet <table> Definition Dynamic Table Row Construction Logic </body> </html></pre>

Word & Excel Velocity Training Template Review

HIGH-LEVEL OVERVIEW OF TEMPLATE STRUCTURE & CODE



```
Generic_Excel_Velocity_Training_Template.vm
1  MIME-Version: 1.0
2  X-DocType: Workbook
3  Content-Type: multipart/related; boundary="-----_NextPart_01D8DFBE.2078AF30"
4
5  #####
6  ##
7  ## Name: Generic Excel Velocity Training Template.vm
8  ##
9  ## Output Type: Excel XLXS
10 ##
11 ## Description/Notes:
12 ##
13 ## Context Sensitive: T/F
14 ##
15 ##
16 ## =====
17 ## MODIFICATION HISTORY
18 ##
19 ## Date      Revision  Person      Comments
20 ## -----
21 ##
22 ##
23 ##
24 ##
25 ## =====
26 ##
27 ## Criteria:
28 ##
29 ##      Type:      Display Name:      Report Global Variable:
30 ##      -----
31 ##
32 ##
33 ##
34 #####
```

Helpful Debugging Tips

ERROR MESSAGES & DEBUGGING



Where to find Velocity Error Messages

JAMA CLOUD HOSTED

- In the event that you install a custom report that has a syntax error present and you run it on the Jama Connect system the report should generate immediately but instead present you with the following error message.

	A	B	C	D	E	F	G	H	I	J	K	
1	The Velocity engine encountered an error in the report											
2												
3	org.apache.velocity.exception.ParseErrorException: Encountered ")" at velocity-engine[line 677, column 21]											
4	Was expecting one of:											

- On the very first line of the error log you will be provided the error summary and line / column number reference where the syntax error was encountered for you to resolve.

Where to find Velocity Error Messages

SELF HOSTED – PRIOR TO JAMA RELEASE 9.7

- If you are hosting a Jama Connect version prior to 9.7 then be aware that if you running an Excel report using the .xlsx format with a syntax error, you will instead be presented with the following error output.

	A	B	C	D	E	F	G	H	I
1	The Velocity engine encountered an error in the report								
2									

- If you encounter this we recommend first adjusting your report installation output format to the legacy .xls file format and re-running the report again to produce the error log output as shown on the prior slide.
- With that you can identify where you need to resolve before adjusting the report installation output format back to .xlsx file format.
- This issue was resolved for all Jama Connect versions 9.7+ and above.

Trouble with Rich Text?

RICH TEXT FIELDS ARE HTML SOURCE DEFINITIONS

- Jama Connect rich text field types are supplied to your custom report as a single block string of HTML definition content viewable using the 'Source' option.

DESCRIPTION:

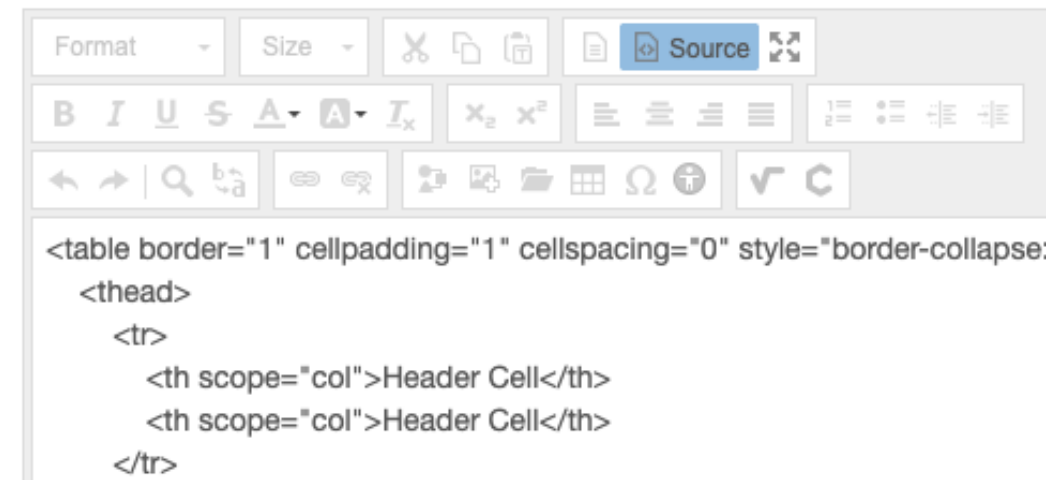
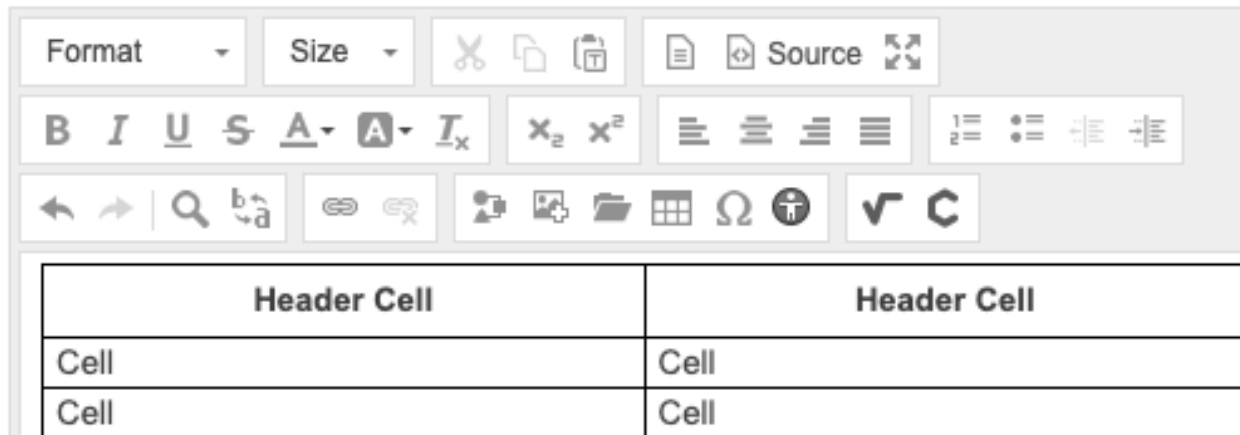


- To ensure we can generate your document effectively such fields are standardly processed by a collection of macros that manage HTML cleaning, styling, table / image resizing, reference handling, etc...
- However, when users paste content from external sources there is a risk of including invalid, fractured, or improper HTML definitions that may not show in the rendered view but are still present in the source definition received by the report and Microsoft Word / Excel.
- While both Jama Connect and our custom report rich text handling macros capture and directly resolve a wide range of such issue content there is still a risk in which you can end up with unexpected outputs

Trouble with Rich Text?

ISSUE OUTPUT RESOLUTIONS

- If you encounter an issue output, do not attempt to modify the rich text field handling macros to handle an invalid HTML definition as this typically results in more issues.
- Custom Reports start with your field definitions in which you can easily resolve the output using either of the options below after navigating to the identified item field:
 1. If you have HTML / CSS experience, you can attempt to resolve the 'Source' HTML definition directly.
 2. Recreate the content using the Jama Rich Text Editor options which will always produce valid HTML definitions that your custom report and or Microsoft Word / Excel can process every time.



EESR Customer Modified Report Policy

In the event that a customer decides to modify a Jama-developed custom report, any resulting defects are not covered under the 3-month defect warranty period provided upon engagement delivery. Customers can contact their Jama Software Consultant or CSM to submit a billable request ticket in order to resolve any caused defects, if required. Customers will be expected to provide their modified report code file to our team or to request the original Jama-developed report be updated with their reporting needs. Any customer-modified reports will be subjected to higher scoping estimates to account for the additional time required to review or reimplement the added code in order to ensure they meet custom reporting standards. All requests for customer-modified Jama-developed reports remain subjected to our team's normal backlog timeline.