# Docs Online – 2022

## UPDATED GUIDE FOR CONNECTING COGNOS TO DOCS ONLINE

Revision 3
Updated 12/13/2022

Michael Webb
CAL POLY POMONA FOUNDATION
MLWEBB@CPP.EDU

## Table of Contents

## Revision List

### Release:

Information

- Initial Release

### Revision 1:

- **CHANGE**: Updated the Paperclip image from .gif to .png in the examples and in all the files in the .zip folder
- **CHANGE**: Fixed Typos in comments within the script and the tutorial documentation
- **CHANGE**: Previous version of FetchControl.js and inline script updated
  - Reports that when the clock image was wiped from the page a broken element appeared on the document.
    - **Fix**: element is removed after no document was found.
- **ADD**: Validation step was added to check if the page had already been iterated through and if the fetch request already added a paperclip icon or wiped the clock icon. This works for Custom control as it holds the previous page node lists in memory.
  - **Reason**: This allows for less Fetch requests to occur and not overwhelm the server you are sending the requests to.
  - **Note**: The Inline Script version of this cannot perform the validation successfully and hold previous page fetch requests in memory. It is best to limit the amount of fetch requests on a single page to not overwhelm the server. This needs to be a value you experiment with and can not be prescribed easily. We don’t go more than 20-30 requests on a page.

### Revision 2:

- **CHANGE**: Installation procedure changed from manual saving files to a Cognos Extension file. This greatly reduces failure points for installation and makes the entire process much easier.
  - This documentation has updated the installation procedure to reflect this change. No more manual saving of images and JavaScript files.
  - The HTML Item Functions for the Classic Viewer/ Legacy script has changed to reference a FetchFunctions.js file that is included in the uploaded extension
  - Language for legacy scripts has been modified to say classic viewer for inline HTML item and Interactive viewer for custom control modules.
  - Verified to work with CA 11.1.7

### Revision 3:

- **MAJOR CHANGE:** Custom Control modules no longer need a Unique ID Data Item. The script has the ability to get a unique id from the document itself. Legacy/Classic Viewer scripts still require a unique ID to work correctly.
- **FIX:** The previous revision had an issue where the FetchControl.js file had a testing script that was not supposed to be used. This version has the intended code and a few minor revisions.
  - This will fix issues where the custom control module throws an error about the *str.replaceAll()* function not working.
- **CHANGE:** There was minor changes to the JavaScript for the custom control that changes where data for validation was stored. If you navigate to the element on the page that the paperclip should be stored the span attribute “*data-status*” is now updated correctly.
- **CHANGE:** Functions within FetchControl.js were cleaned up for readability and stability.

# Docs Online

## Why this Update?

The Central Square default method for accessing Docs Online was very useful for legacy reporting in Cognos 10.x and early 11.0.x but for speed and reliability there was a few details that were missed in the **"SUGA2016 - Cognos Reporting with Docs Online"** and the "**Docs Online"** updated scripting. These scripts prevented us from utilizing advanced features within Cognos for advanced reporting and interactivity while feeling like a downgrade from CDD Reporting features.

A key problem was that these scripts use a ***synchronous*** XMLHttpRequest() as an inline HTML Item. This meant that as a report was being generated, the Document Object Model (DOM) or "Webpage", had to complete all the individual "GET" requests within a page before a user could navigate to a subsequent page, where this process would repeat. Now for attachments for PO's this wasn't a huge deal but for many different kinds of workflows with a variety of attachment definitions this slows down the query and ultimately the rendering of a HTML report page.

For our use case, this could make opening a multiple page report with 20 lines per page incredibly slow. For that one page alone, it would consist of 20 "GET" requests that each had to wait for the previous one to complete before beginning and blocks a user from interacting with any part of the rendered page until it completes. This came out to about 1-3 seconds per request and a full minute-long process for each page rendered.

For our clients, this made the script made Cognos difficult to justify using in place of CDD Reports and a decreased trust in the Cognos reporting system.

The second reason we needed to update the script was that since IBM Cognos 11.0.x to the present 11.1.7 long term revision IBM shifted the way it drew the DOM from their classic viewer which required unorganized and unoptimized inline scripts to the Interactive viewer which uses **Custom Control Modules**.

Modules were a feature that allowed all the JavaScript scripts in a page to be bundled and executed independently from each other which helps segregate variables from interfering with each other, scripts could be debugged more reliably, and updates made in a single file rather than updating the same inline script in multiple reports. It also can allow parameters and data to be passed to the script from a JSON configuration field that is a feature of the Custom Control properties.

The future of modern browsers and web design is using this to increase speed and enhance UI experiences. IBM moved in this direction with Cognos so that we could add advanced interactive features to the report like collapsible regions, interactive visualizations like D3.js, Map Box API, and much more.

I have created two versions of the DOL script so that you can replace your legacy scripts with an asynchronous version in either a full Interactive report or a limited interactive report with the Classic Viewer (Legacy).

## What is a Custom Control?

Well, we have rewritten the original script from Central Square to be a custom control which runs with a **promise based asynchronous Cognos Custom Control**. This means that as a user opens a report the "GET" request is happening in the background and automatically updates the DOM (report) with a hyperlink to the intended Attached Document **without interfering with the loading and navigation of the report.**

This is a huge step forward for speed and reliability as it allows users to the freedom to do the work they need to do. This tutorial document will reference similar concepts from previous documentation from Central Square while showing the method for implementing this new method.

## References for Custom Controls

You can find a demo of Custom Controls here at:
Overview: Scriptable reports (11.0.4+) - YouTube

They also have documentation for scriptable reports and how to write modules that will work for the newer scriptable reports here:
https://public.dhe.ibm.com/software/data/sw-library/cognos/mobile/scriptable_reports/index.html

JavaScript_setup_instructions.pdf (ibm.com)

Lastly, they have a package of Extended Sample Scripts here for reference:
Extended Samples for Cognos Analytics (ibm.com)

There is also a free extension for Cognos named CogBox built by PMSquare which adds the ability to copy and paste custom control modules and templates pretty easily between reports. They have lots of videos showing how it works and is a helpful tool but is not necessary for this tutorial. You can find it referenced here along with a multitude of custom control tutorials:

- CogBox for Cognos Analytics — PMsquare
- Analytics Blog — PMsquare

## Run with full interactivity? "Yes"

In the past, for us to run JavaScript files in the HTML Items we usually set the "Run with full interactivity" to "No" but for our new custom control it **requires that we switch this to "Yes".** This will break any legacy HTML Items in your report, whether it is found on the prompt page or on the report itself and cause an error. Those are only supported in the classic viewer. We recommend removing any inline HTML items you have in your report that do not fit IBM's new standard for well formed HTML if you transition to the Interactive Viewer.

**NOTE**: **If you want an asynchronous script that does not use the custom control feature** I included an adapted version of the script that runs asynchronously that you can use in limited interactivity mode. Although in the long term it may be useful to update your reports to the newer format and convert your current inline HTML scripts to custom controls as IBM has standardized it going forward. It is very similar to how we setup the custom control except you set the Full interactivity to "No", and you use 3 HTML Items. Text files with each HTML Item specifications, and an XML Report Specification is also included to copy into your own report. This legacy script does not save previous page fetch requests responses in memory so every time you change pages it will re-run the fetch for that page.

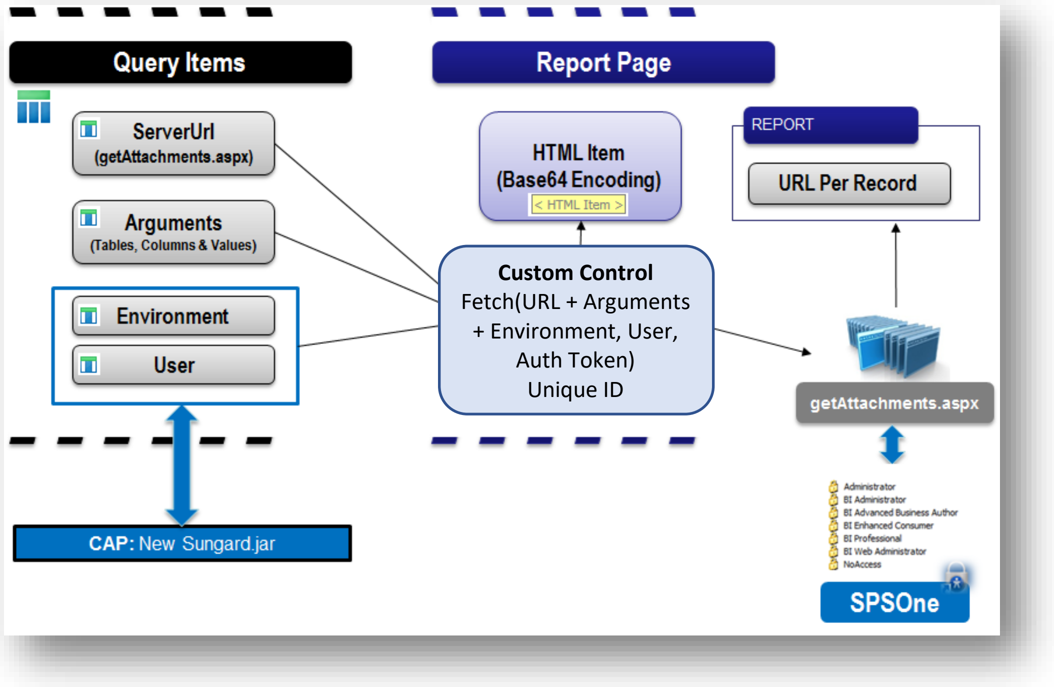
## Concept

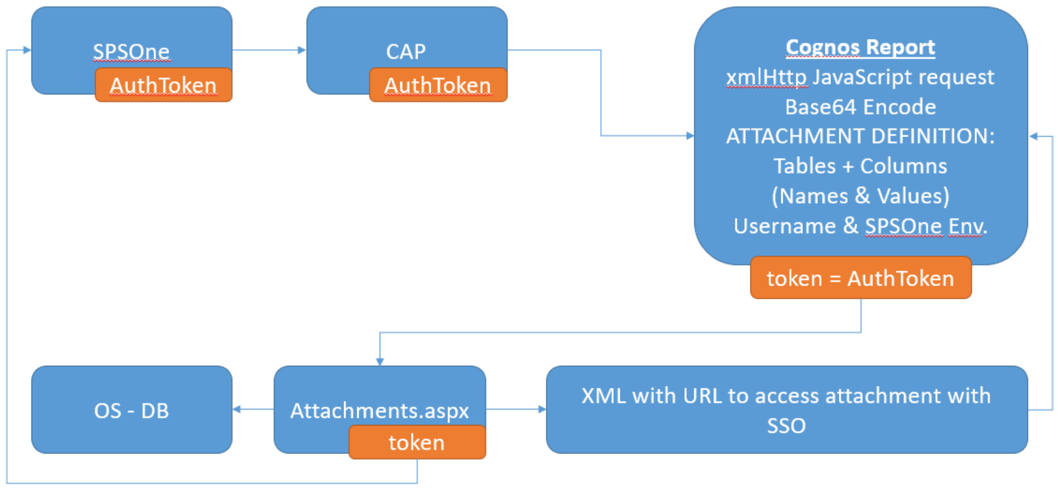
***This section is adapted from previous documentations but is still useful to understand what is happening in the background of the script***

The purpose of this instruction is to automate a drill through hyperlink in a Cognos report to a document kept in the client documents repository. That repository is called Docs Online (DOL). There are several steps involved in successfully setting up a DOL Link.

There is an xml document attached that provides a sample of how it is all set up successfully – the best thing to do is paste this sample on the client system to use as a model to test and it will have the SQL and functions needed for each field and HTML tag.

## Drill-Through to Documents Online functionality from a Cognos report





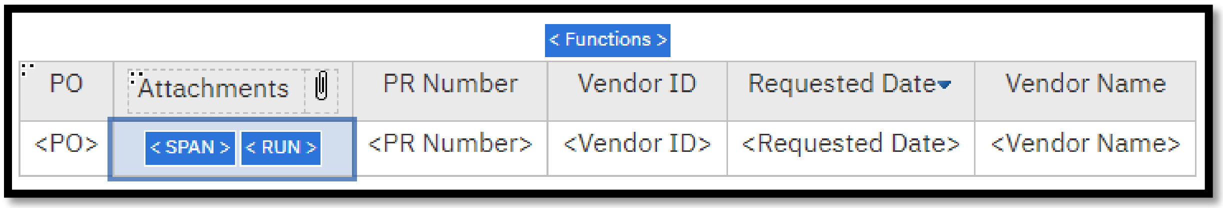
## Prerequisites

- **Cognos Report Studio:** Advance reporting techniques
- **FE Attachment Definitions**: General technical understanding
    - Knowledge of attachment definition table and column items tied to image type to drill-to.
- **FE Documents Online:** Basic Understanding
- **JavaScript:** Basic understanding
- **Latest CAP Files**
- **Host Name:** Location in the network for **getAttachments.aspx** file

*e.g. http://**<hostname>**/Finance/Documents/getAttachments.aspx*

### Files Included

- **DocsOnline.zip**
    - **FetchControl.js**: This is the custom control JavaScript file.
    - **FetchFunction.js**: This is a classic viewer (legacy Inline HTML) script.
    - **Images**: Paperclip.png and Clock.png
    - **Spec.json**: This informs Cognos the files associated with this extension.
        - Do not change the contents of this file unless you understand how to create and modify extensions for Cognos.
- **SPAN HTML Item:** This is the text you must place in an HTML Item within the repeating column that you want the icon hyperlink to appear. This passes the query information to the script
- **PO Report XML Spec:**
    - You can copy the text in this file and copy it to report clipboard to run our demo report. You may need to fix the query dependencies on a package but it's pretty plug and play.
    - **NOTE**: There are two versions, a custom control version and a legacy version.
- **Inline HTML files (legacy style):**
    - "Functions – HTML Item (Text)": copy this file into an HTML Item at the top of the body of your report.



< Functions >

| PO | Attachments | PR Number | Vendor ID | Requested Date | Vendor Name |
|---|---|---|---|---|---|
| <PO> | < SPAN > < RUN > | <PR Number> | <Vendor ID> | <Requested Date> | <Vendor Name> |

- "RUN HTML Item (Report Expression)": This file needs to be copied into an HTML Item with a Source Type: "Report Expression" and placed in a repeating element within a column. This function is used to run the script per each line.

## What is Documents online

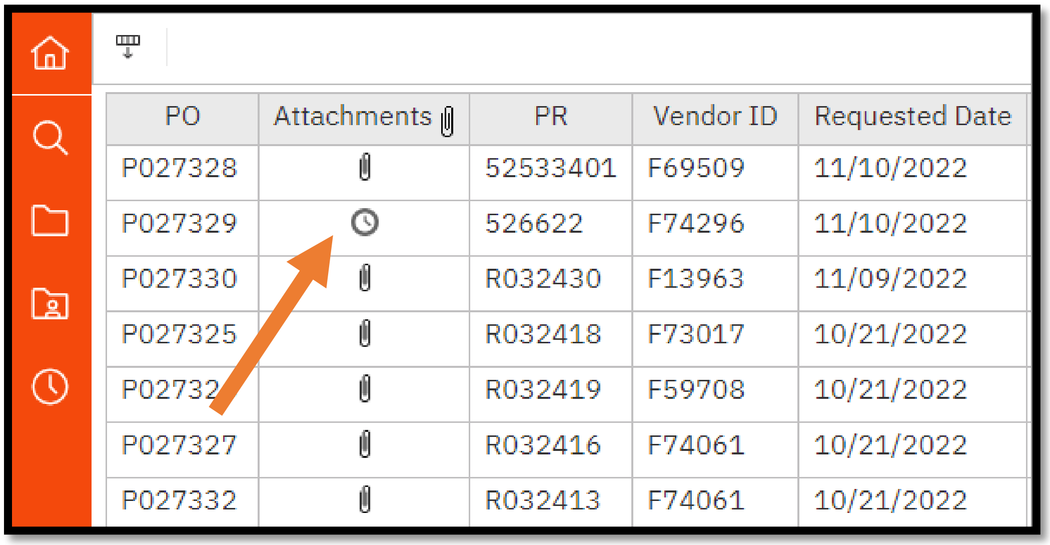
Documents Online provides you with the ability to store images in the database and relate those images to information in Finance.

## Attachment Definitions

The Attachment Definition controls which Tables and Columns will be used to link a Document to Finance Data. This can be as simple as a link to a single table or a more complex link referred to as "Document Progression".

## What does a Link to Docs Online Look Like?

Report output:



| PO | Attachments | PR | Vendor ID | Requested Date |
|---|---|---|---|---|
| P027328 | 📎 | 52533401 | F69509 | 11/10/2022 |
| P027329 | 🕓 | 526622 | F74296 | 11/10/2022 |
| P027330 | 📎 | R032430 | F13963 | 11/09/2022 |
| P027325 | 📎 | R032418 | F73017 | 10/21/2022 |
| P02732 | 📎 | R032419 | F59708 | 10/21/2022 |
| P027327 | 📎 | R032416 | F74061 | 10/21/2022 |
| P027332 | 📎 | R032413 | F74061 | 10/21/2022 |

Click on the paperclip link to open attachments from Documents Online.

## First things first

Go to Cognos create a new blank report with a FE package with all the data needed for Attachment definitions.

Open Finance Enterprise (FE) and locate the where you would get a particular document, in this case it is we will link a PO. The screen POUPPR has the link to the Docs online for PO's.

- Open a Document.
- When the PDF Opens look at the web-address, you are going to need a portion of it for the steps to link docs, so keep that doc open for now.

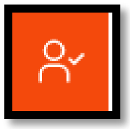
Now you have everything you will need to build your docs online link from your report!

## Steps to build a Docs Online Link

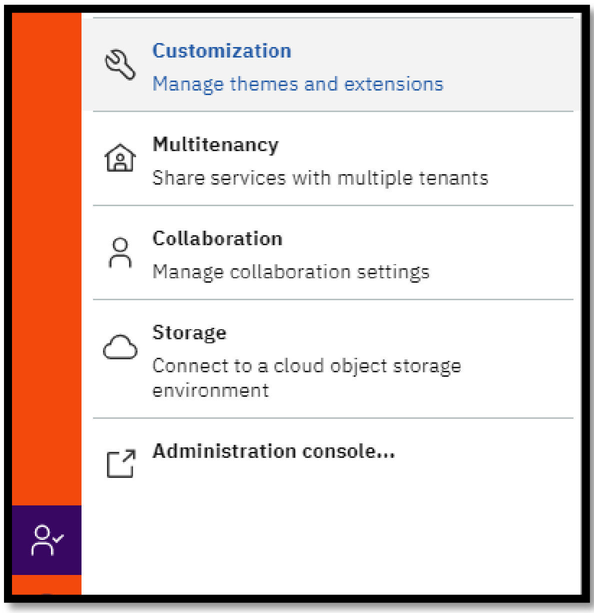
There are 8 Query Calculation/ Data Items to build in the query and you must save 3 files on the Cognos server.  They all need to be spelled consistently or else they will not work.

### Upload the **DocsOnline.zip** Extension

- To upload an extension you may need your cognos administrator permissions.
- Click on the 'Manage' Icon on your Cognos AppBar



- Click on 'Customization'



**Customization**
Manage themes and extensions

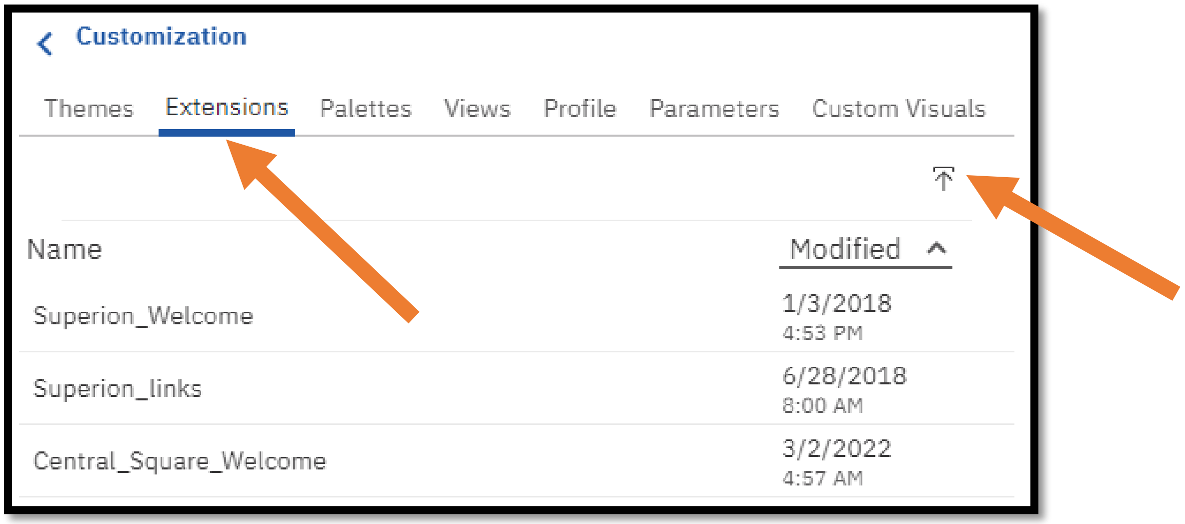
**Multitenancy**
Share services with multiple tenants

**Collaboration**
Manage collaboration settings

**Storage**
Connect to a cloud object storage environment

**Administration console...**

- Then click on the 'Extensions' Tab at the top of the window and click the upload icon

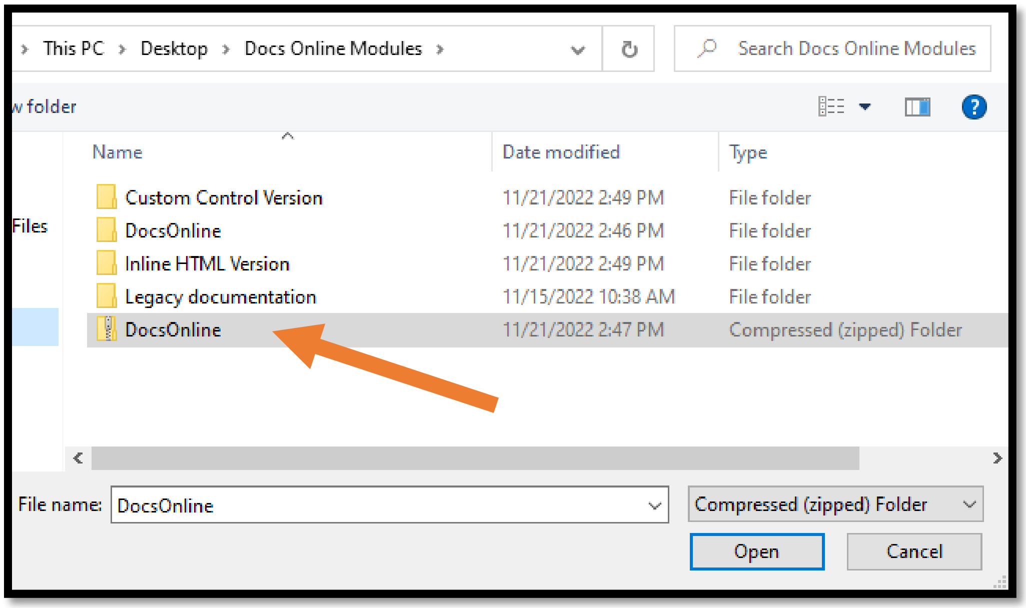


**Customization**

Themes | Extensions | Palettes | Views | Profile | Parameters | Custom Visuals

| Name | Modified |
|---|---|
| Superion_Welcome | 1/3/2018 4:53 PM |
| Superion_links | 6/28/2018 8:00 AM |
| Central_Square_Welcome | 3/2/2022 4:57 AM |

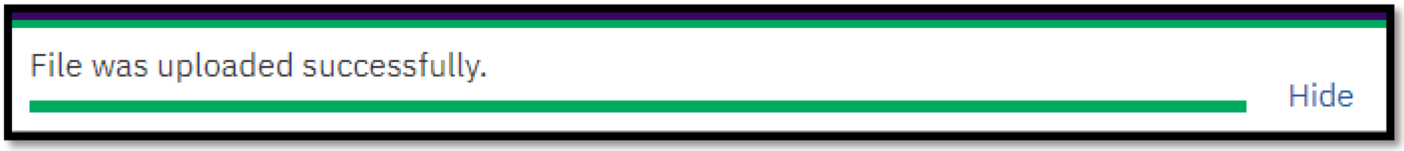
- Then navigate to where you downloaded the Custom Control folder and click on the zipped folder DocsOnline.zip and 'Open'


This PC › Desktop › Docs Online Modules ›
Search Docs Online Modules
w folder
Files
Open
Cancel

| Name | Date modified | Type |
|---|---|---|
| Custom Control Version | 11/21/2022 2:49 PM | File folder |
| DocsOnline | 11/21/2022 2:46 PM | File folder |
| Inline HTML Version | 11/21/2022 2:49 PM | File folder |
| Legacy documentation | 11/15/2022 10:38 AM | File folder |
| DocsOnline | 11/21/2022 2:47 PM | Compressed (zipped) Folder |

File name: DocsOnline
Compressed (zipped) Folder

- Once successfully uploaded you should receive a small notification that looks like this:

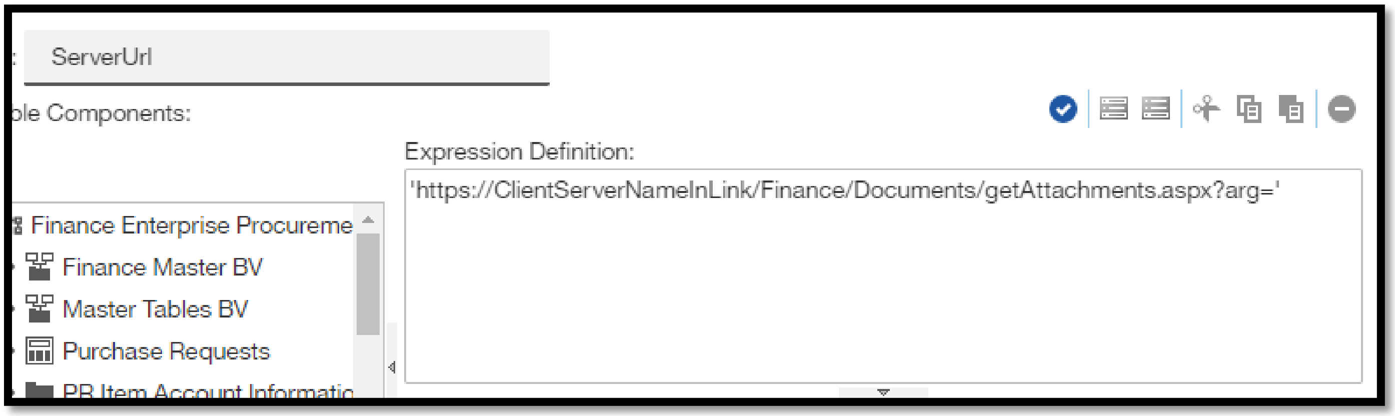

File was uploaded successfully.

You have successfully installed all the files server side needed to start your report.

## Server URL

- Create a new Data Item - Query Calculation
- Name it '**ServerUrl'**
- Go to the previously opened (PDF) document
- Start the expression off with a  single quote (') and end with a single quote (')
- Extract the Server portion of the web address (https://....../Finance/Documents) and paste it into the Expression box.
- The Expression Box should look like :

```
'https://......./Finance/Documents/getAttachments.aspx?arg='
```


ServerUrl

ble Components:

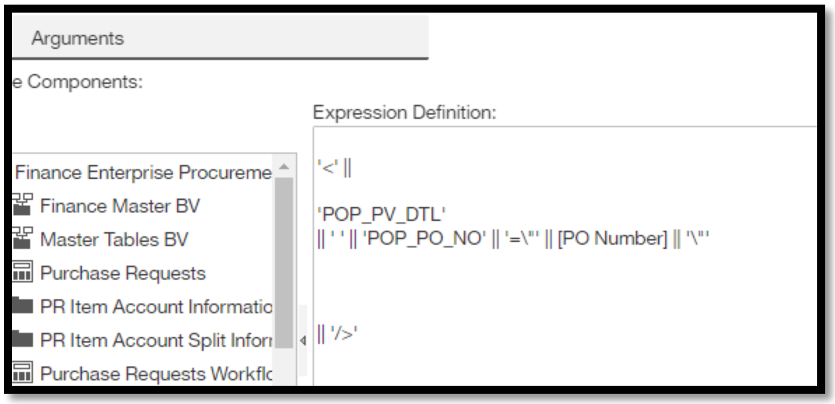
Expression Definition:

```
'https://ClientServerNameInLink/Finance/Documents/getAttachments.aspx?arg='
```

## Arguments

The Arguments Data Item is to translate the database table field to what it is named in the query

- Create a new Data Calculation
- Name it '**Arguments'**
- Identify the *table name* and *column name* from the attachment definition.
- in this case for the PO Number it is table: **POP_PV_DTL** and column: **POP_PO_NO**
- And the field is named **PO Number** in the report query (note in the below example you are bringing in the field from the query hence the [braces] around PO Number


Arguments

e Components:

Expression Definition:

```
'<' ||

'POP_PV_DTL'
|| ' ' || 'POP_PO_NO' || '=\"' || [PO Number] || '\"'


|| '/>'
```

Finance Enterprise Procureme
Finance Master BV
Master Tables BV
Purchase Requests
PR Item Account Informatio
PR Item Account Split Infor
Purchase Requests Workflo

```
'<' ||

'POP_PV_DTL'
|| ' ' || 'POP_PR_NO' || '=\"' || [PR Number] ||
'\"'

|| '/>'
```

NOTE: Multiple Fields can be used to match your attachment definitions

## Environment

- Create a new data calculation
- Name it '**Environment'**
- Paste the SQL exactly as it is and close it

```
'&env=' || #sq($account.parameters.ActiveEnvironment)#
```

## User

- Create a new data calculation
- Name it '**User'**
- Paste the SQL exactly as it is and close it

```
'&env=' || #sq($account.parameters.ActiveEnvironment)#
```

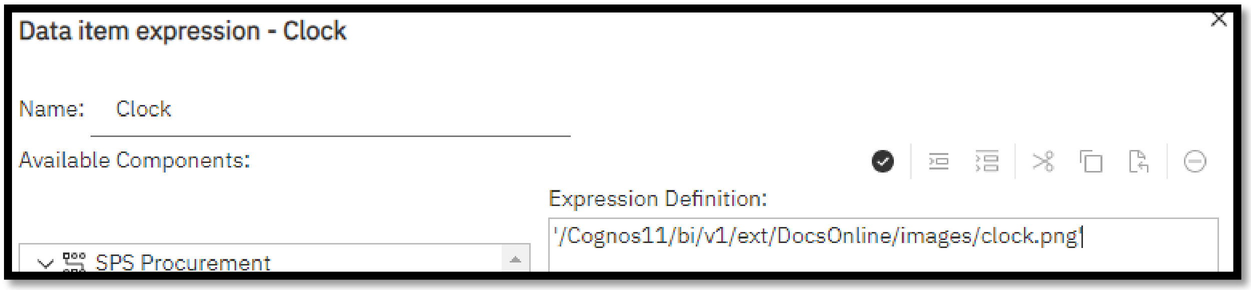
## Token

- Create a new data calculation
- Name it **Token**
- Paste the SQL exactly as it is and close it

```
# sq(URLEncode($account.parameters.AuthToken)) #
```

## Clock and Paperclip

- First you must copy and save two images to a folder on the Cognos server that we can reference.
- Depending on where you save it most commonly the path will look *similar* to this:
    - '/Cognos11/bi/v1/ext/DocsOnline/images/clock.png'
    - '/Cognos11/bi/v1/ext/DocsOnline/images/paperclip.gif'
- Create a new data calculation
- Name it '**Clock'** and **'Paperclip',** respectively.
-

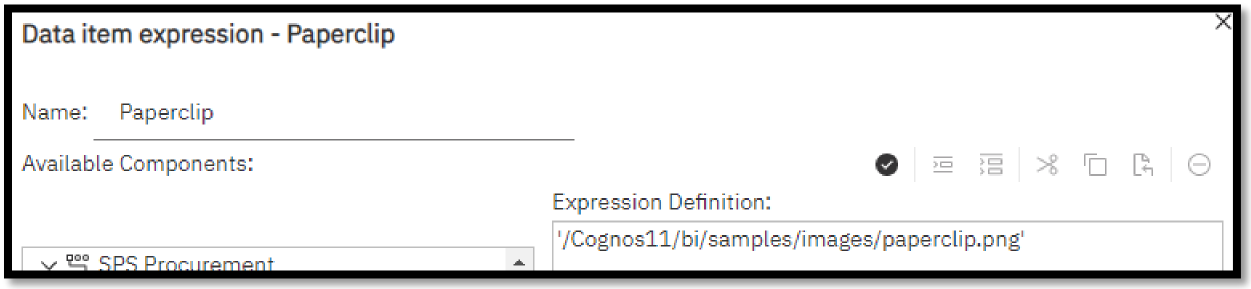

Data item expression - Clock

Name: Clock

Available Components:

Expression Definition:

'/Cognos11/bi/v1/ext/DocsOnline/images/clock.png'


Data item expression - Paperclip

Name: Paperclip

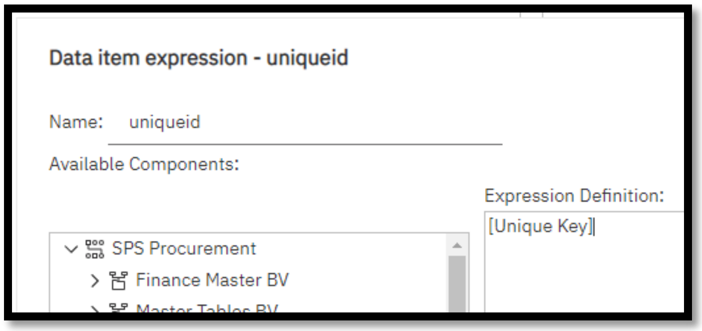
Available Components:

Expression Definition:

'/Cognos11/bi/samples/images/paperclip.png'

## Unique ID

- This is required for Classic Viewer scripts and no longer used in Rev 3 of the Docs Online extension for custom controls.
    - **NOTE:** The Custom Control does not need this step but if you choose to use it make sure to **add** the *data-unid = "[Unique Id]"* from the SPAN HTML Item.
- Create a new data calculation
- Name it '**uniqueid'**
- **This is where you must be creative and find a way to have a unique id or unique key for each line in the report.** I recommend creating or using a unique key that is associated with table row you are referencing.


Data item expression - uniqueid

Name: uniqueid

Available Components:

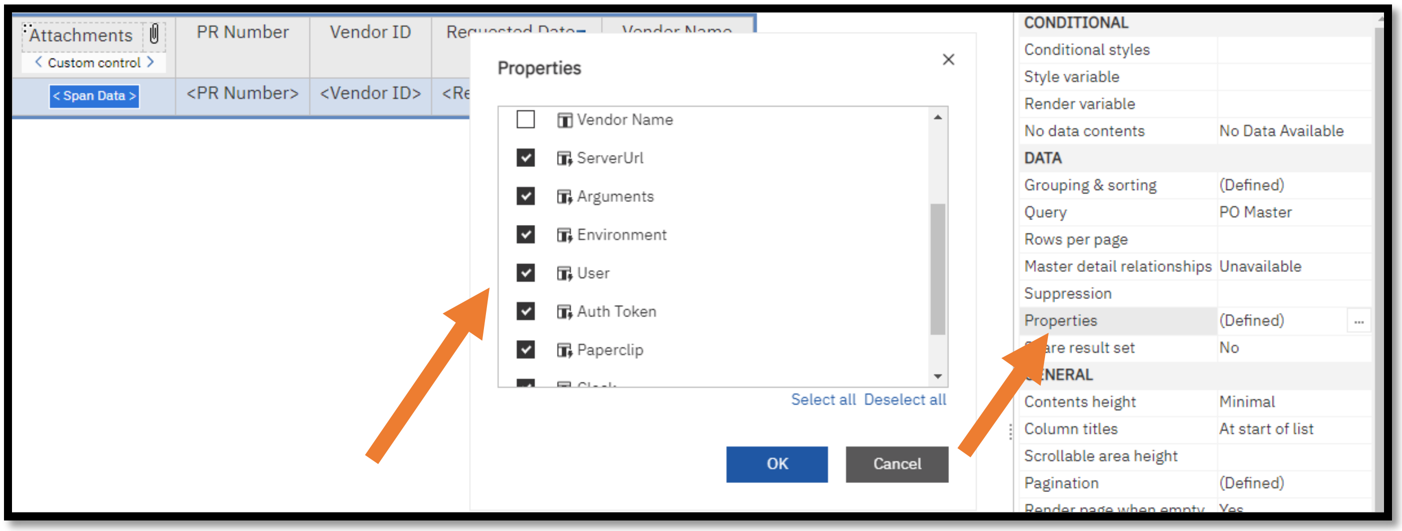
Expression Definition:

[Unique Key]

## Add HTML Span and Custom Control to the Report Page

There is only one HTML Item to add to the report page and one Custom Control to add to the page.

- Unlock the report
- Make sure that all the additional query items we created are added to the list properties



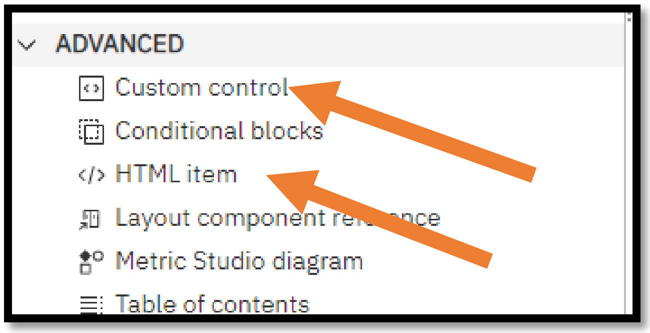
| Attachments | PR Number | Vendor ID | Requested Date | Vendor Name |
|---|---|---|---|---|
| < Custom control > | | | | |
| < Span Data > | <PR Number> | <Vendor ID> | <Re | |

Properties

- [ ] Vendor Name
- [x] ServerUrl
- [x] Arguments
- [x] Environment
- [x] User
- [x] Auth Token
- [x] Paperclip
- [x] Clock

| | |
|---|---|
| **CONDITIONAL** | |
| Conditional styles | |
| Style variable | |
| Render variable | |
| No data contents | No Data Available |
| **DATA** | |
| Grouping & sorting | (Defined) |
| Query | PO Master |
| Rows per page | |
| Master detail relationships | Unavailable |
| Suppression | |
| Properties | (Defined) |
| Share result set | No |
| **GENERAL** | |
| Contents height | Minimal |
| Column titles | At start of list |
| Scrollable area height | |
| Pagination | (Defined) |
| Render page when empty | Yes |

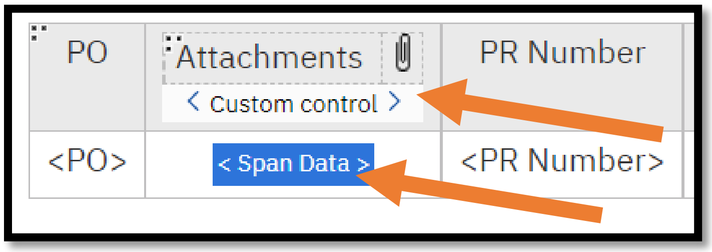
- Go to the Toolbox Icon then to the advanced section
- Drag an HTML Item and position it within the column and line you would like to have the icon appear.



- We named our HTML Item “Span Data”.
- Paste the Script within the HTML Item

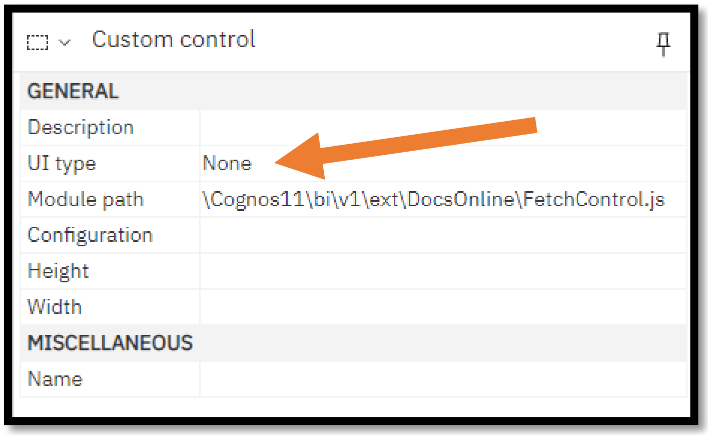
```
'<span name="attachments" data-token="'+[Auth Token]+'" data-arg="'+URLEncode([Arguments])+'" data-
user="'+URLEncode([User])+'" data-env="'+URLEncode([Environment])+'" data-server="'+URLEncode([ServerUrl])+'" data-
clock="'+URLEncode([Clock])+'" data-paperclip="'+URLEncode([Paperclip])+'" data-status = "new" data-font="16"/>'
```

- Drag a Custom Control and place it anywhere on the list that will be iterated through. I recommend you place it in the head as seen in the picture below:



| PO | Attachments | PR Number |
|---|---|---|
| | < Custom control > | |
| <PO> | < Span Data > | <PR Number> |

- **IMPORTANT**: In the Custom Control Settings, Set "UI type" to "none"
- Set the module path to '\Cognos11\bi\v1\ext\DocsOnline\FetchControl.js'



Custom control

| GENERAL | |
|---|---|
| Description | |
| UI type | None |
| Module path | \Cognos11\bi\v1\ext\DocsOnline\FetchControl.js |
| Configuration | |
| Height | |
| Width | |
| **MISCELLANEOUS** | |
| Name | |

- This will allow our report to pass the data item values to our custom control as a value hidden in the document
- Run the Report!

## Classic Viewer (Legacy) Report Scripts

The difference for reports in classic viewer mode is minor compared to the old documentation.

- All the Query Calculations remain the same
- Make sure Report Setting ***Run with Full Interactivity*** is set to "No"
- Report Page needs 3 HTML Items. The contents of the HTML items can be found in the included Inline HTML text files
    - **SPAN** (Source Type= Report Expression):

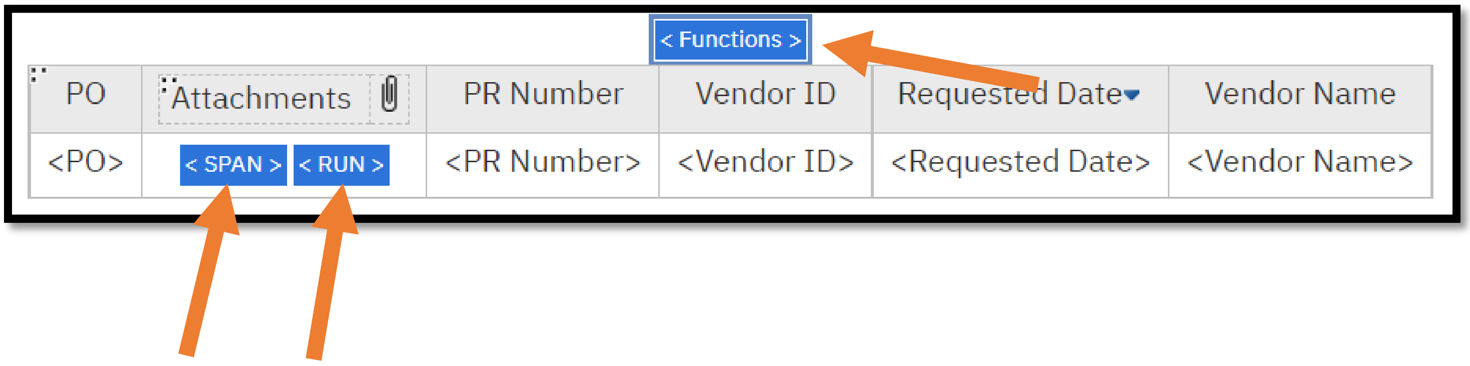
```
'<span name="attachments" data-token="'+[Auth Token]+'" data-unid="'+URLEncode([uniqueid])+'" data-
arg="'+URLEncode([Arguments])+'" data-user="'+URLEncode([User])+'" data-env="'+URLEncode([Environment])+'" data-
server="'+URLEncode([ServerUrl])+'" data-clock="'+URLEncode([Clock])+'" data-paperclip="'+URLEncode([Paperclip])+'" data-status =
"new" data-font="16"/>'
```

    - **RUN** (Source Type= Report Expression):

```
'<script type"javascript" defer> checkForAttachment() </script>'
```

    - **Functions** (Source Type= Text):

```
<script src = "/Cognos11/bi/v1/ext/DocsOnline/FetchFunctions.js"></script>
```

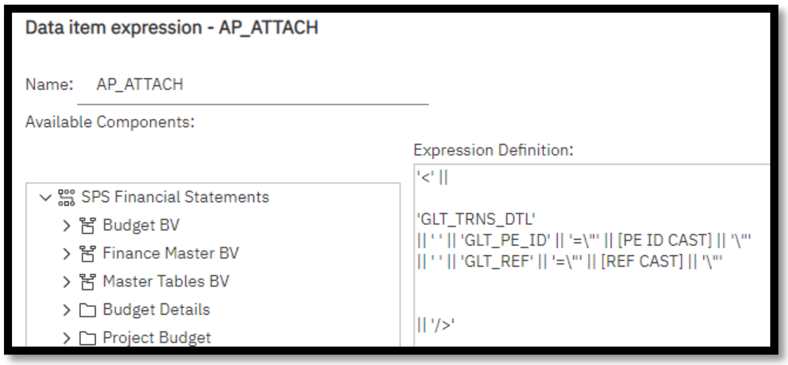


## Troubleshooting and Formatting

- **Icon Font Size:** If you want to change the size of the icons that are populated then open the Span Data HTML Item and change the value for data-font=”16” to your desired size.
    - If you choose to change icons to a different image you might need to check the script itself for the length and width dimensions. The clock is square but the paperclip image width is calculated by your data-font size value divided by 2.
- **Data Lengths:** It is possible that if you use the “trim()” function on any of the data items you are passing to the script that it will not work. The Documents Online service ( “…/getAttachments.aspx”), references the exact data, data type, and data length in their respective tables. That means if your [PR Number] data item is 8 characters in the reference table but you pass a shortened 6 character string to Doc Online, it will return a null. There is an easy way to fix this:
    - Create a query calculation to cast the exact string length of each value. For example:
        - *‘right(trim([PR Number])+space(8-character_length([PR Number])),8)’*
    - Add this calculation to the list properties and use it as the data item to pass to the Custom Control Span.
- Things to remember to double check if the script is not working:
    - Check that the Query Items and the items in the URL and the List HTML Item are all spelled the same
    - Ensure the Query name has been changed to the query name you put the new Items in; in both the List HTML Item and the URL query item.
- If you get a server error
    - Ensure you got the entire server information from the PDF that popped up when you opened a Doc from FE (for Items relating to POs use the screen POUPPR)
    - It should look something like this https://ClientServerNameInLink/Finance/Documents/
- To Diagnose any data value issues, you can press F12 to open developer console on your browser.
    - The node list that we iterate through will appear in the console and you can look within it to the sub-level named “datasets” and this is where the values from our SPAN HTML Item are stored. This is the same data that Docs Online will see in a fetch request.

## For Developers

- The Document Online service can return a full hyperlink in HTML if you would like instead of an XML response. This requires that you add "&html=true" to the end of the Fetch URL link. It will automatically return a HTML element in this format:
    - ‘<a href=”………..”target=”_blank”>Files(s)<a/>’
- The getAttachments.aspx returns “” and “ “ for no document or if the string sent doesn’t exist. If you see a ‘+’ symbol in the encoded Argument in the GET URL then you have an encoding issue. Check to see that you aren’t passing any special characters and that it is decoding the strings correctly.
        - I have purposely encoded all the strings sent to the script so users don’t have to mess with the JavaScript but it is still possible that a double \\ or // can break it. This is a hard problem to troubleshoot so always check that what is going into the custom control module is what you want the Docs Online service to see.
- This script can still be cleaned up with making more discrete functions and other tweaks so feel free to adapt it for your uses. We have a multitude of attachment definitions for our system so I built the XML argument for each attachment definitions and have a CASE statement that chooses the appropriate definition for particular subsystems and workflow types. Shown Below:


Data item expression - AP_ATTACH

Name: AP_ATTACH

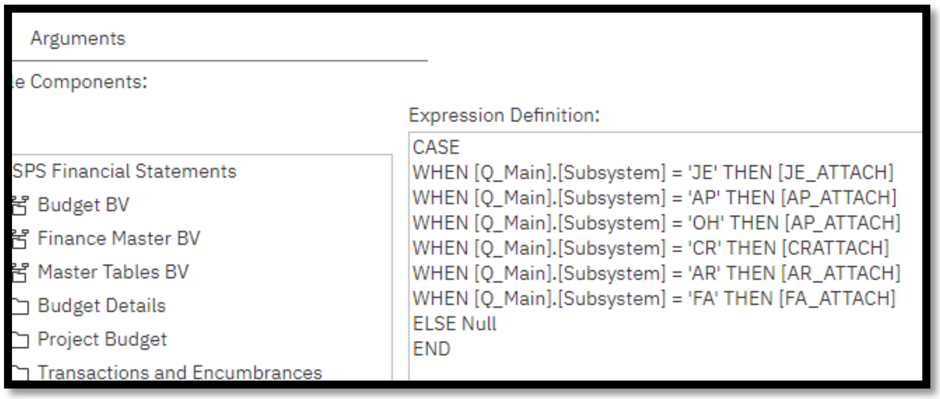
Available Components:

Expression Definition:

```
'<' ||

'GLT_TRNS_DTL'
|| ' ' || 'GLT_PE_ID' || '=\"' || [PE ID CAST] || '\"'
|| ' ' || 'GLT_REF' || '=\"' || [REF CAST] || '\"'


|| '/>'
```


Arguments

e Components:

Expression Definition:

```
CASE
WHEN [Q_Main].[Subsystem] = 'JE' THEN [JE_ATTACH]
WHEN [Q_Main].[Subsystem] = 'AP' THEN [AP_ATTACH]
WHEN [Q_Main].[Subsystem] = 'OH' THEN [AP_ATTACH]
WHEN [Q_Main].[Subsystem] = 'CR' THEN [CRATTACH]
WHEN [Q_Main].[Subsystem] = 'AR' THEN [AR_ATTACH]
WHEN [Q_Main].[Subsystem] = 'FA' THEN [FA_ATTACH]
ELSE Null
END
```