

Item Reference Configuration

Overview

Item Reference Configuration provides the ability to assign JavaScript Methods that get invoked when a referenced Item is created through the process of inserting an instance of an Item Document Element (see Item Elements) or via file Drag and Drop (as is the case with Image Files). These JavaScript Methods provide the logic to pre-populate Property values on the newly created Item and validate Property values added/updated by the user in the Item Form.

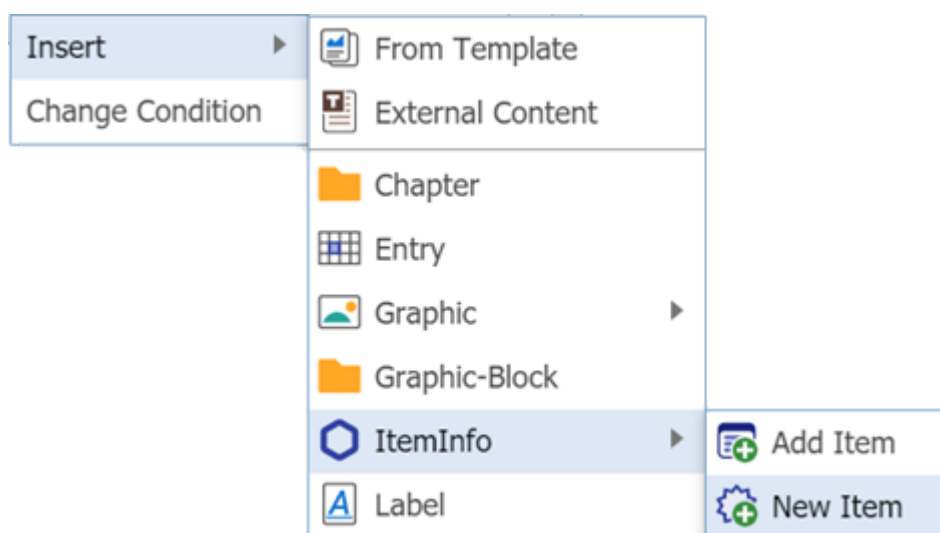
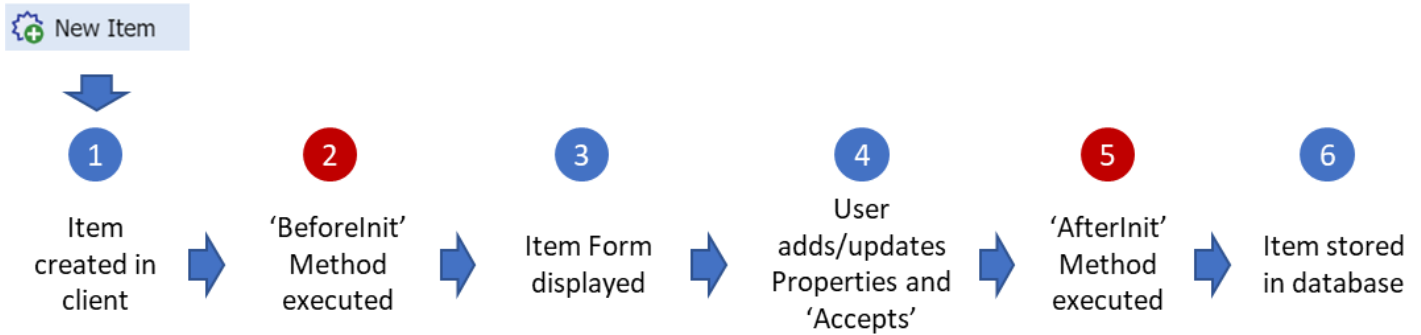


Figure: Generating a New Referenced Item

As described in Section Item Elements, Item Document Elements contain a link between a Document Element in a Technical Document and an Item (Business Object) maintained in Innovator. Often, Properties from the associated Item (or related Items) are used to populate content within the Document. In addition to selecting an existing Item, the Technical Document Editor allows the creation of the referenced Item at the time the Item Document Element is placed. Doing so often requires input from the author to populate Properties on the Item before the record is created in the Database. When configuring a Document Element, it is possible to assign a JavaScript Method that will be executed before the Item Form is presented to the author and after the Form is 'Accepted' to generate the Item in the database and establish the reference between it and the Technical Document. The process of creating a referenced Item is as follows:



1. The Authors selects the 'New Item' context menu for the Item Document Element. The system will generate a new referenced Item in the client. Note the same process can be initiated when dragging an Image file from the file system on to the canvas of a Technical Document that is opened for Edit. In this case, an `tp_Image` ItemType will be created and referenced.
2. The configured JavaScript Method is called to pre-populate Properties on the Item
3. The associated Item Form is presented to the Author/User
4. The author visually validates the content and/or updates as required and selects the 'Accept' Button
5. The configured JavaScript Method is called to validate the Properties on the Item
6. The Item is stored in the database

The reference to the newly created Item is established with the Technical Document once the Technical Document is saved. In the case where an Image File has been dragged onto the Editor canvas, an Image Document Element, of the appropriate type, will be created and added to the document with the image displayed.

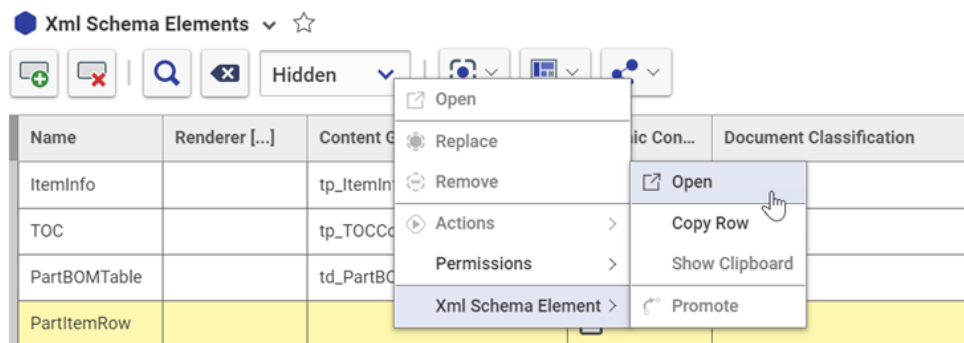
Configuration

The configuration options for Item Reference Configuration allow:

- The identification of an alternate Form to be used for Property Updates
- A Method used to pre-populate Item Properties
- A Method used to validate Item Properties
- The option to bypass the display of the Item Form

This configuration is specified using JSON for the XML Schema Element associated with the Item Document Element.





Xml Schema Element

Name

PartItemRow

Renderer

Generator Parameters

Editor Parameters

Figure: XML Schema Element Form

The **Editor Parameters** input form is configured for displaying JSON content. Syntactical errors will be displayed if the text content doesn't conform to JSON formatting. To set the Item Reference Configuration, a single JSON object is used with the following properties:

```
{
  "itemCreation": {
    "formId": "<id of Form Item to use as alternate>",
    "onBeforeInit": "<beforeInit Method Name>",
    "onAfterInit": "<afterInit Method Name>",
    "initializerType": "<initializer type>",
    "fileAccept": "<true/false>",
    "allowedMIMETypes": "<[comma-delimited array of mime type text]>",
  }
}
```

Important



Additional configuration options may be added in the future. The use of JSON provides a flexible means of specifying configuration options without the need for ItemType modifications in future releases

Table: Item Reference Configuration Options

JSON Property	Description
<code>itemCreation</code>	JSON Object Name for Item Creation options.
<code>formId</code>	(Optional): Identifies an alternate Form to use for specifying Item Properties. The default Form is used for the referenced ItemType if this is not used.
<code>onBeforeInit</code>	(Optional): Specifies the name of a JavaScript client Method to be called prior to the Form displayed to the User.
<code>onAfterInit</code>	(Optional): Specifies the name of a JavaScript client Method to be called after the Form displayed to the User.
<code>initializerType</code>	(Optional): Accepts values: ' base ' and ' formDialog ' when used for non-Image ItemType references. ' formDialog ' is the default and is used when a Form should be displayed to the author/user. ' base ' is used to suppress the Form display. When ' base ' is specified the before <i>and</i> after initialization Method will still be executed if they are specified. When this Property is used for Image Document Elements for file Drag and Drop support, a value of 'coreLocalFileImage' should be used to generate a tp_Image ItemType.
<code>fileAccept</code>	Used for Drag and Drop of Image Files to auto-create Image Document Elements. When 'true', the system will accept image file drag/drop onto the Editor to auto-create an Image Document Element. 'false' (or not included) to not allow image creation via Drag and Drop
<code>allowedMIMETypes</code>	(Optional): list of image MIME Types that are allowed. If not included, all image types supported by the system will be allowed.

onBeforeInit Method

Name
 td_PartReferenceConfigBeforeInit |

Comment
 Method to populate a new Part Item when genera |



```

1  this.setProperty("item_number", "123abc");
2  this.setProperty("product_info", "E1271B3D9AFF4ED38DAF42CC3BE2FBA7");
3  this.setProperty("name", "Test");
4
5
6  return inDom;

```



Figure: onBeforeInit Method Example

The purpose of the **onBeforeInit** Method is solely to pre-populate selected Properties of the Item being created. It is meant to provide a convenient mechanism to ensure consistent data for Items and should execute quickly, without excessive memory or processing. The **onBeforeInit** Method is a JavaScript Method that will be passed two parameters – **inDom** and **inArgs** – and must return the **inDom** input object. The context object of the Method is the **Item** being created. Thus ' **this** ' refers to the Item Object and the standard IOM Item API applies. The **inDom** parameter represents the XML content of the Item being created also. As such, methods like **aras.setItemProperty(inDom, ...)** will work as well. The **inArgs** parameter is an object with two properties: **documentItem** and **viewModel** . **documentItem** contains the XML representation of the Technical Document Item in which the Document Element is being created. This object can be used to retrieve information about the Technical Document Item. Likewise, methods like **aras.getItemProperty(inArgs.documentItem, ...)** will work. The **viewModel** object represents the API for the Technical Document Editor.

Important

The **viewModel** object is not part of a supported API. Methods/Properties can be modified without notice in subsequent releases and is to be used at the customer's risk.

onAfterInit Method

Name	td_PartReferenceConfigAfterInit	Comment	
 Client-side JavaScript			
<pre>1 var prop = this.getProperty("name"); 2 3 if(prop == null prop.length == 0) 4 { 5 aras.AlertError("something bad happened!"); 6 return null; 7 } 8 else 9 return inDom;</pre>			



Figure: onBeforeInit Method Example

The purpose of the **onAfterInit** Method is solely to validate selected Properties of the Item being created. It is meant to provide a convenient mechanism to ensure consistent data for Items and should execute quickly, without excessive memory or processing. The **onAfterInit** Method is a JavaScript Method that will be passed two parameters – **inDom** and **inArgs** – and must return the **inDom** input object. The context object of the Method is the **Item** being created. Thus ' **this** ' refers to the Item Object and the standard IOM Item API applies. The **inDom** parameter represents the XML content of the Item being created also. As such, methods like **aras.getItemProperty(inDom, ...)** will work as well. The **inArgs** parameter is an object with two properties: **documentItem** and **viewModel** . **documentItem** contains the XML representation of the Technical Document Item in which the Document Element is being created. This object can be used to retrieve information about the Technical Document Item. Likewise, methods like **aras.getItemProperty(inArgs.documentItem, ...)** will work. The **viewModel** object represents the API for the Technical Document Editor.

Important

The **viewModel** object is not part of a supported API. Methods/Properties can be modified without notice in subsequent releases and is to be used at the customer's risk.

If there is a validation error, an alert/message should be displayed, and **null** should be returned. When ' **base** ' is used for the **initializerType** in the configuration, returning **null** will prevent the Item from being created.

