



Template Builder User Guide

Version 1.2



Template Builder



Copyright © 2019 Vizrt. All rights reserved.

No part of this software, documentation or publication may be reproduced, transcribed, stored in a retrieval system, translated into any language, computer language, or transmitted in any form or by any means, electronically, mechanically, magnetically, optically, chemically, photocopied, manually, or otherwise, without prior written permission from Vizrt. Vizrt specifically retains title to all Vizrt software. This software is supplied under a license agreement and may only be installed, used or copied in accordance to that agreement.

Disclaimer

Vizrt provides this publication “as is” without warranty of any kind, either expressed or implied. This publication may contain technical inaccuracies or typographical errors. While every precaution has been taken in the preparation of this document to ensure that it contains accurate and up-to-date information, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained in this document. Vizrt’s policy is one of continual development, so the content of this document is periodically subject to be modified without notice. These changes will be incorporated in new editions of the publication. Vizrt may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time. Vizrt may have patents or pending patent applications covering subject matters in this document. The furnishing of this document does not give you any license to these patents.

Technical Support

For technical support and the latest news of upgrades, documentation, and related products, visit the Vizrt web site at www.vizrt.com.

Created on

2019/04/26

Contents

1	Introduction.....	5
1.1	Workflow	5
1.2	Document Structure	6
1.3	Related Documents	6
1.4	Customer Feedback and Suggestions	6
1.5	Customer Support Requests	7
1.6	Before Submitting a Support Request	7
1.7	Submitting a Support Request	7
1.7.1	Content of a Support Request	7
1.7.2	To submit a Support Request:	8
2	Setup	9
2.1	Preview Server	9
2.1.1	Setting Database Parameters for Preview Server	9
2.2	Template Builder and Mist	9
3	Getting Started	10
3.1	Open a Template	10
3.2	Main User Interface	10
3.3	Open	11
3.4	Save.....	12
3.5	Restore	12
3.6	About	12
4	Template Preparation.....	13
4.1	Creating a Template	13
4.2	Managing Concepts and Variants	13
5	The Template Builder Window	15
5.1	Model	15
5.1.1	Field Tree	15
5.1.2	Field Properties	16
5.2	Settings	21
5.2.1	Duration	21
5.2.2	Track.....	21
5.2.3	Title generation	22
5.3	Data Entry.....	23

5.3.1	Manual	23
5.3.2	Choose from list	23
5.3.3	Enable feed browser/Parent feed browser	28
5.4	The HTML Panel	30
5.4.1	Browser Caching	31
5.4.2	Creating HTML templates.....	31
6	The Fill In Form.....	42
7	The Graphics Preview Window	43
7.1	Auto-refresh	44
8	Keyboard Shortcuts.....	45
8.1	Graphics Preview.....	45

1 Introduction

Template Builder is a web application that allows journalists and content creators to edit and build customized templates based on existing templates that have been generated and organized using [Viz Pilot's](#) Template Wizard.

The templates can be customized using different functionalities described in this guide. The major functionality in Template Builder is the possibility to add custom HTML panels to templates which gives the user full control over the template by using custom scripting and logic.



1.1 Workflow

The basic workflow from graphics design to a working template is as follows:

- Scenes are made in Viz Artist. Multiple scenes can make up one template, categorized under different variants.
- Objects in the scene, that the graphics designer wants to make editable in the finished template, are given the appropriate control plugins.
- Templates are made in Template Wizard, after importing the scene(s) over.
- These templates are saved within the Pilot system and are available to the Template Builder.
- In Template Builder a template can be opened, edited/customized and previewed.
- The template is saved back into the Pilot system and is made available to the newsroom and control room systems.

- A journalist, editor or other content creators, fill the graphics template with content. This can include: text, numbers, images, videos, websites and maps. The template is saved into the Viz Pilot system creating a data element that is added to the newsroom story.
- The Viz Pilot operator monitors the newsroom playlist (rundown) and plays the graphics on-air at the correct time in the broadcast.

**Note:**

The changes made to a template in Template Builder will not be available in the template in Template Wizard

1.2 Document Structure

This User Guide is divided into the following chapters:

- [Introduction](#)
- [Setup](#)
- [Getting Started](#)
- [Template Preparation](#)
- [The Template Builder Window](#)
- [The Fill In Form](#)
- [The Graphics Preview Window](#)

1.3 Related Documents

The templates customized in Template Builder can be used by other Viz products like Viz Pilot Edge, Viz Story and Viz Multiplay.

For more information about all of the Vizrt products, visit:

- www.vizrt.com
- [Vizrt Documentation Center](#)
- [Vizrt Training Center](#)
- [Vizrt Forum](#)

1.4 Customer Feedback And Suggestions

We encourage suggestions and feedback about our products and documentation.

To give feedback and, or suggestions, please identify your local Vizrt customer support team at www.vizrt.com.

1. Click on **Contact** (top of page).
2. The Vizrt office which is nearest to your location will be shown, or select from the list of Vizrt offices.
3. Click on the **Contact** button for the office you want.

4. Complete the required details in the window that open
5. A Vizrt representative will contact you as soon as possible.

⚠ Note:
If this message is for Customer Support, and there is a Support Contract in place, then click on the 'For support requests, please visit our support portal' link in the message window.

1.5 Customer Support Requests

Support Requests are supported by Vizrt if customers have a valid Service Agreement in operation. Customers who do not have a Service Agreement and would like to set up a Service Agreement should contact their regional sales representative (see [Customer Feedback and Suggestions](#)).

When submitting a Support Request, relevant and correct information should be given to Vizrt Support, to make sure that Vizrt Support can give the quickest and best solution to your Support Request.

This section contains the following topics:

- Before Submitting a Support Request
- Submitting a Support Request

1.6 Before Submitting A Support Request

Before a Support Request is submitted make sure that you:

Read:

- The relevant User Guide or Guides
- The release notes

and Check:

- That the system is configured correctly
 - That you have the specified hardware, tested and recommended versions
- Always refer to your Vizrt Service Level Agreement document.

1.7 Submitting A Support Request

When completing a Support Request, add as much information as possible.

1.7.1 Content of a Support Request

The report should contain information about these topics:

- **Problem description:** Include a good description of what the problem is and how to reproduce it. Specify your workflow. Remember to use simple English.

- **Screenshots and illustrations:** Use these to simplify the message. These are extremely useful for Vizrt Support.
- **Software configuration:** Add exact versions of software used. This is extremely important information. The version information is available in the log.
- **System log files:** Send the system log file. You can save your log file in the *link to Logs pane*.
- **System locale:** Specify the Region and Language settings of the system.
- **Hardware configuration:** Add exact versions of hardware used, especially for Viz Engine.

Optional:

- **System setup:** Describe differences in the installation, if any, from the recommended setup.
- **System Network:** Add a description of how the network, bandwidth, routers, and switches are configured.
Always refer to your Vizrt Service Level Agreement document.

1.7.2 To submit a Support Request:

1. On the www.vizrt.com page, click on Support.
2. Click on **Report a case**.
3. Click on **LOG IN** to login to the Customer and Partner portal.
4. At the top of the Case Management page, click on **Report a Case**.
5. In the online form complete the required minimum information (shown by a red asterisk) and click **SAVE**.
6. In the saved Support Case that opens, complete the various text boxes and upload any required documents, files, etc. (see [Content of a Support Request](#)).
To **track the status** of open support tickets, login to the Customer and Partner portal. Add information or communicate about the cases directly with the support team.

2 Setup

Template Builder opens as a web application in your default browser.

The URL to access Template Builder, if hosted on the Pilot Data Server, is:

```
http://pds-host-name:8177/templatebuilder
```

2.1 Preview Server

The Preview Server manages one or several Viz Engines providing frames for thumbnails and snapshots in an ongoing preview process.

The Preview Server must be configured in the Pilot Data Server:

2.1.1 Setting Database Parameters for Preview Server

1. See how To access the Pilot Data Server Web Interface in the Pilot Data Server section in the [Pilot User Guide](#).
2. Click the **Settings** link
3. Select the **preview_server_uri** setting, and add the parameter for the machine you installed the Preview Server on (i.e. `http://<hostname>:21098`)
4. Click **Save**

This will give all applications with a connection to the database access to the Preview Server.

2.2 Template Builder And Mist

To make resource sharing with other products like Viz Pilot Edge and Viz Story simpler, or if you need HTTPS support, it is recommended to install **Mist** and let that serve out all the products.

The URL to access Template Builder if served out by Mist is:

```
http://hostname/templatebuilder/
```

3 Getting Started

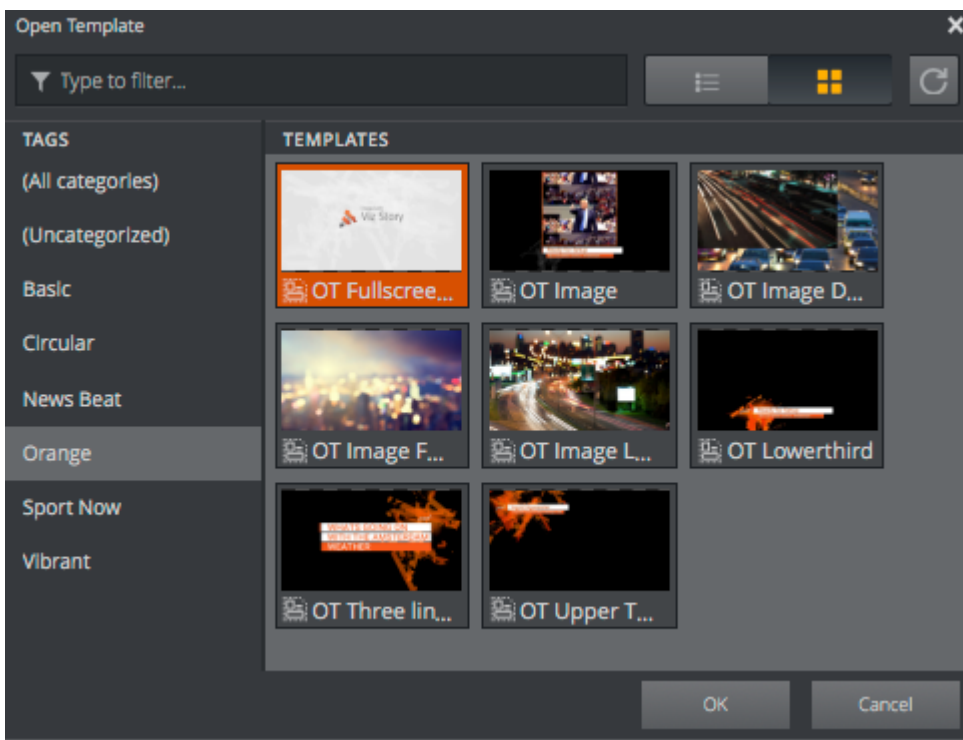
This section will explain the main user interface of Template Builder and a short introduction on how to use it.

3.1 Open A Template

Template Builder is a tool for editing and customizing *existing* templates. Click **Open**, or **Ctrl + O**, to open the dialog showing the existing templates available within the Pilot system.

In the **Open Template** dialog, use **TAGS** to filter on templates. The template search can also be narrowed down by searching for the template name in the **Type to filter...** field at the top of the dialog.

Select a template and click **OK**, or double-click one, to open it in the Template Builder user interface.

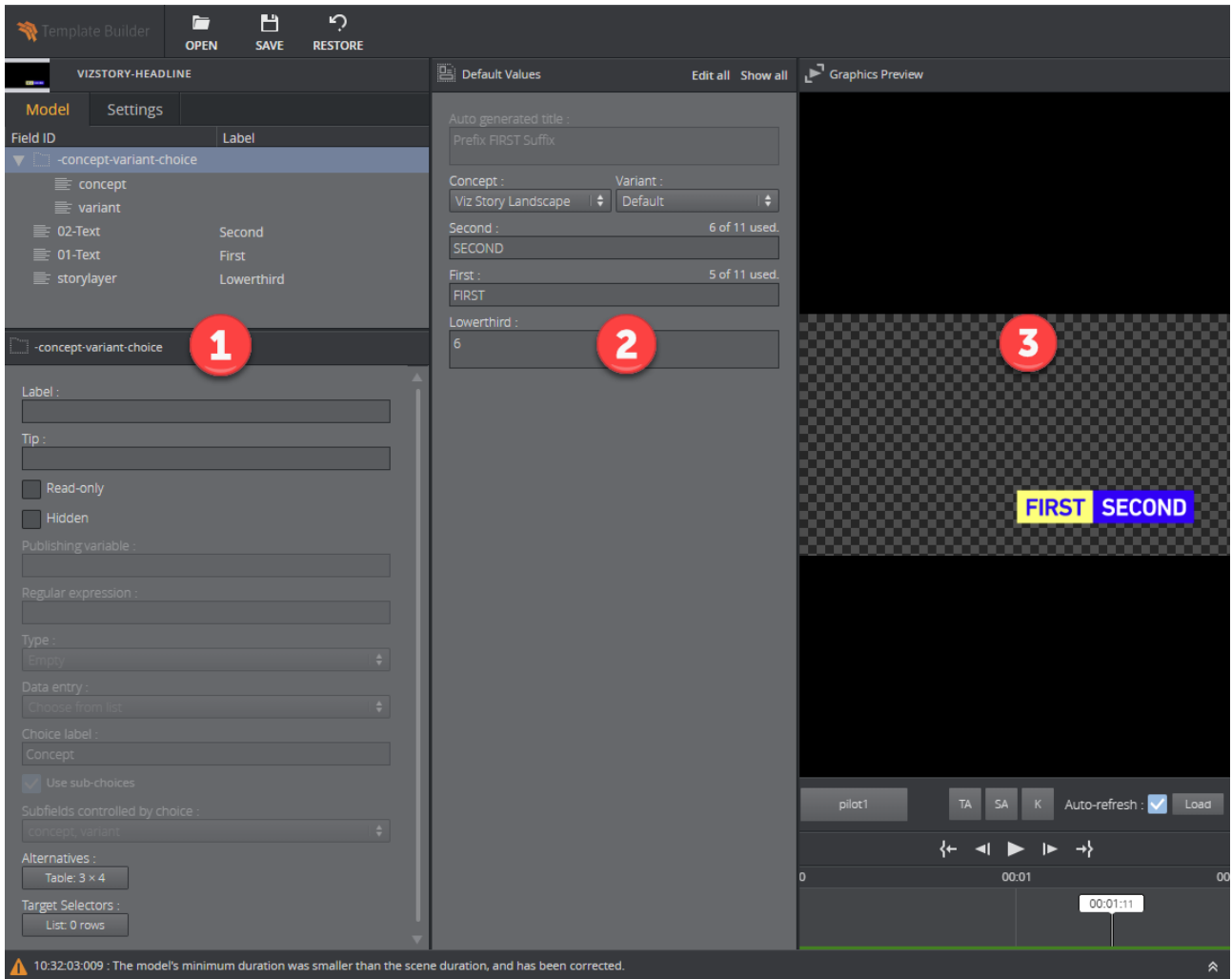


3.2 Main User Interface

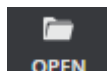
The user interface is divided into three main windows.

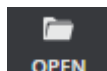
- (1) [The Template Builder Window](#)
- (2) [The Fill In Form](#)
- (3) [The Graphics Preview Window](#)

The Graphics Preview Window (3) provides snapshots of the graphics with content. The preview snapshots provide the user with an idea of how the graphics will look like when played on air and are updated as changes are being made in the Fill In Form (2). In the Fill In Form the user fills in/ edits content for the graphics template. The field information, exposed fields, and layout in the Fill In Form can be customized by the user in the Template Builder Window (1).

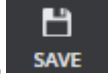



3.3 Open



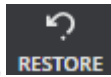
The **Open** button, , is located in the toolbar at the top of the Template Builder window. Clicking it or pressing **CTRL + O** opens the **Open Template** dialog allowing opening another existing template available within the Pilot system. If there are unsaved changes in the template already open, Template Builder will ask whether these changes should be saved before opening a new template.

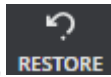
3.4 Save



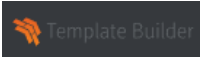
The save button, , is located next to the open button. Clicking it or pressing **CTRL + S** saves the changes made to an open template. The updated template will overwrite the existing template in the Pilot Data Server.

3.5 Restore



The restore button, , will undo all changes since the last save.

3.6 About

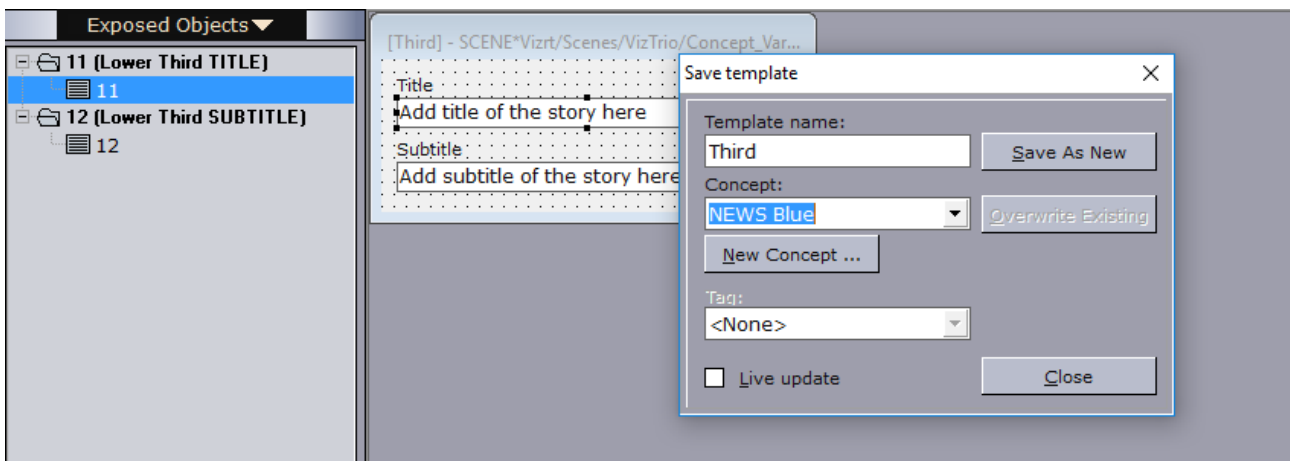
Clicking the Template Builder logo  in the top left corner of the application takes the user to the about page.

Here is information about software version, and links to local documentation and to third-party licenses.

4 Template Preparation

The current version of Template Builder relies on templates being generated and organized using Viz Pilot's Template Wizard. The following procedures will give you a quick introduction on how to create and manage a template. For more information on how to work with Template Wizard, please see the Template Wizard section in the [Viz Pilot User Guide](#).

4.1 Creating A Template

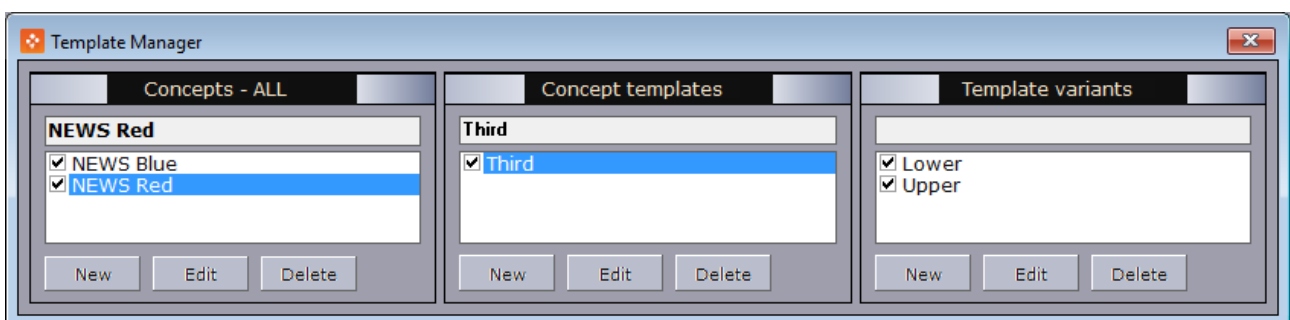


The following procedure explains how to create and save a template.

To create a template:

1. Start Template Wizard
2. Connect to your Viz Engine
3. On the File menu, select Wizard
4. Drag and drop the scene to the Selected Scenes pane (right)
5. Click Finish
6. On the File menu, select Save (Ctrl+S)
7. Enter a Template name
8. Select a Concept name (or create a new)
9. Click Save as New

4.2 Managing Concepts And Variants



The following procedures explain how to organize a template for use with more than one concept and variant, for example, a Blue and Red concept with a lower and upper third graphic. As a prerequisite, all scenes should have the same type of data in order to use the same template.

To manage concepts:

1. On the Tools menu, select Template Manager
2. Under the Concepts - All pane, click New
3. Enter a name for the new concept and save it
4. Open the other concept
5. Drag and drop the template from the old to the new concept (and confirm the operation)
6. Open the new concept
7. Edit the template by selecting the correct Scene folder for your new concept
8. Save and confirm the operation

To manage variants:

1. On the Tools menu, select Template Manager
2. Select your concept, then the template
3. Under the Template variants pane, click New
4. Enter a Description of the variant
5. Select the Scene name that is a variant of the other
6. Save and confirm the operation

5 The Template Builder Window

The Template Builder window is located to the left in the application and is where all the customization of the template is carried out.

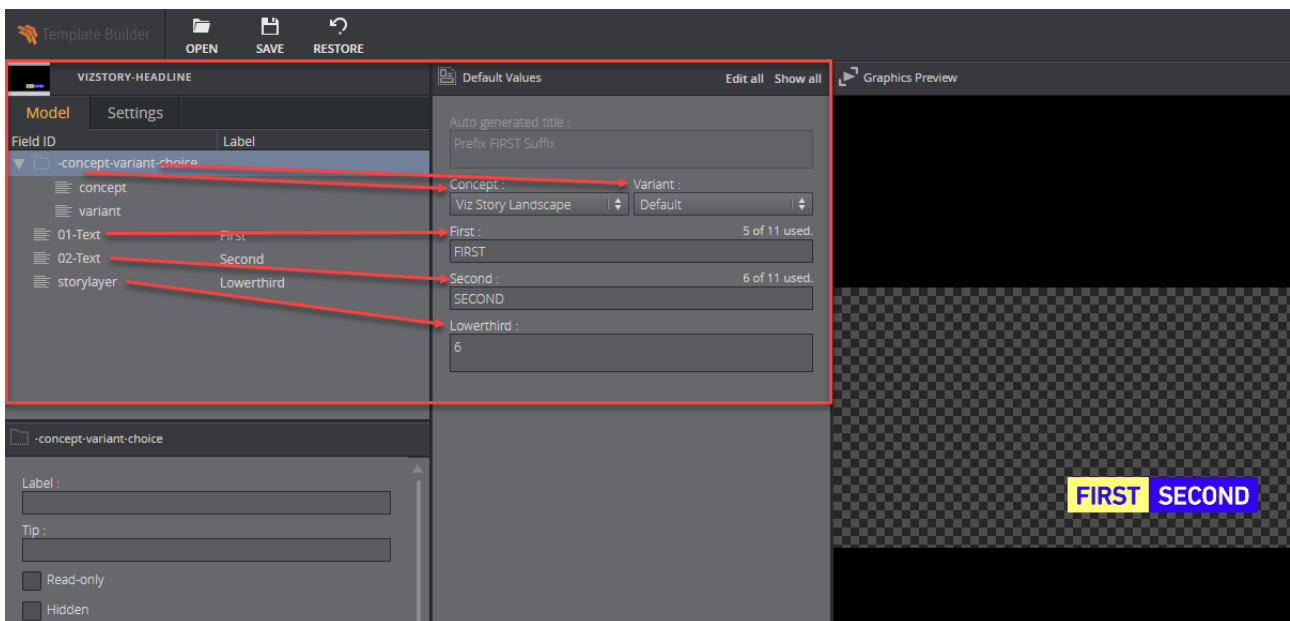
It consists of two tabs:

- [Model](#)
- [Settings](#)

5.1 Model

5.1.1 Field Tree

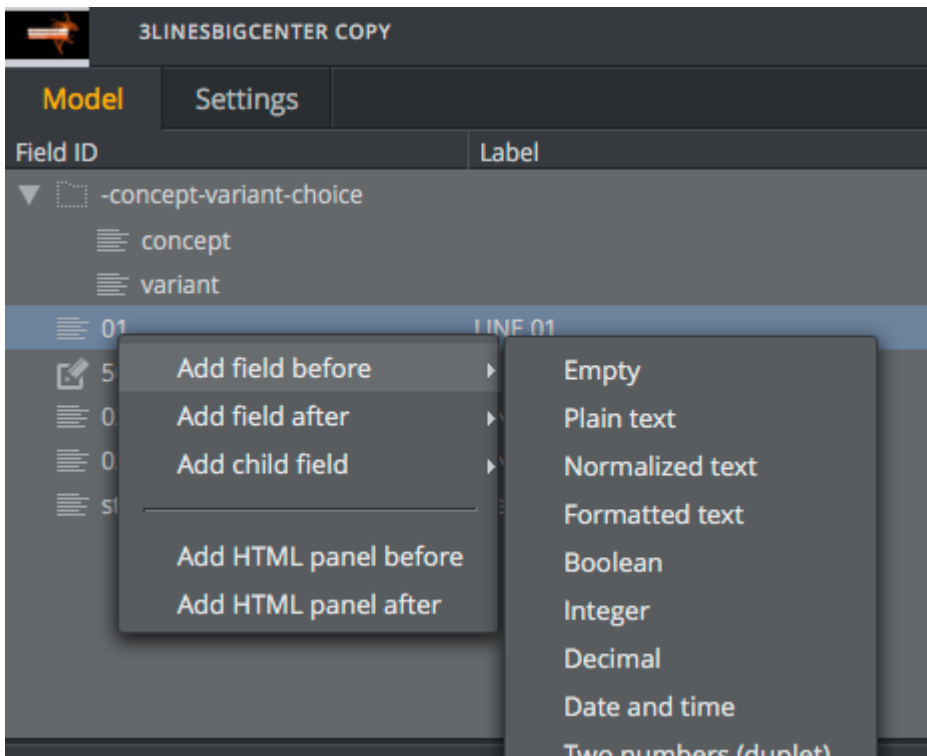
Each line in the Field Tree displays the field's ID and label and often corresponds to a value in the Fill In Form, see the red arrows in the figure below. The associated icon indicates the [Type](#) of the field.



The listing of fields can be rearranged by drag-and-drop within the Field tree.

Right-clicking a field, a list of choices appears. From here additional fields and [HTML panels](#) can be added.

The Fill In Form will immediately update upon any change.



Info

Only fields created in Template Builder can be deleted and given a new ID.

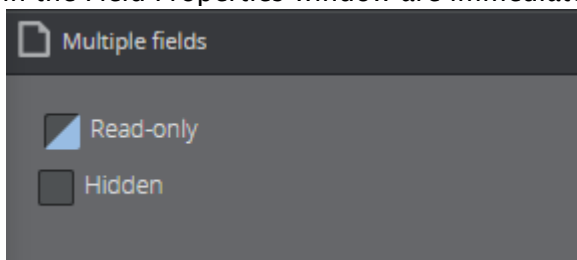
Multi-selection

It is possible to multi-select in the Field Tree. Additional fields may be selected by **CTRL + click**. It is not possible to move or rename multiple fields at the same time, but multiple fields can be deleted and some of their properties can also be changed from the **Field Properties** window.

5.1.2 Field Properties

The **Field Properties** window is located below the **Field Tree** window. It displays the properties of a selected field(s) in the Field Tree.

- **Multi-selection** - If several fields are selected in the Field Tree, a subset of the field properties for the selected fields are displayed. If the selected fields have different field property values, the Field Properties window displays a multiple values state. Changes made in the Field Properties window are immediately applied to all the selected fields.



- **Single-selection** - All properties for the selected field are displayed. Note that the set of properties depend on the **type** of the field.

03

Label :
Omo

Tip :
omo

Read-only

Hidden

Regular expression :

Type :
Integer

Minimum :

Maximum :

Data entry :
Choose from list

Choice label :

Use sub-choices

Alternatives :
Table: 1 x 3

Label

Specifies the label of the field in the Fill-In Form.

Tip

A tooltip text can be entered to give more information about the field.

Read-only

Selecting the **Read-only** checkbox will keep the field visible, but grayed out in the Fill-In Form.

Hidden

Selecting the **Hidden** checkbox will hide the field in the Fill-In Form.

Regular expression










In this field, the user can specify a regular expression which will define constraints of the value in the field.










Type


The type of content allowed in the field in the Fill-In Form is set using the drop-down list under **Type**. Depending on the type selected different sub-choices become available which is specified in the table below.

There are two main field type categories, scalar and list. Fields with all types except the list type are referred to as scalar fields. Fields using the list type are referred to as list fields.

The following types are available:

Type	Icon	Description
Empty		Will make the field unavailable. Typically used as a container for other fields.
Plain text		The field is a text field. Max length: Set the maximum number of characters allowed in the field.
Normalized text		Normalized text. Max length: Set the maximum number of characters allowed in the field.
Formatted text		A text that allows formatting. Max length: Set the maximum number of characters allowed in the field. Single-line: Check this box to specify that the rich-text editor should only allow one line of text.
Boolean		Creates a checkbox which has two states; true and false.
Integer		The field is an integer field. Minimum: Set the minimum value allowed in the field. Maximum: Set the maximum value allowed in the field.
Decimal		The field allows decimal numbers. Minimum: Set the minimum value allowed in the field. Maximum: Set the maximum value allowed in the field.
Date and time		Use the Date Chooser in the Fill-In Form to select date and time in this field.
Two numbers (duplet)		The field allows two numbers (decimal numbers are allowed). Minimum: Set the minimum value allowed for both numbers. Maximum: Set the maximum value allowed for both numbers.

Type	Icon	Description
Three numbers (triplet)		The field allows three numbers (decimal numbers are allowed). Minimum: Set the minimum value allowed for all three numbers. Maximum: Set the maximum value allowed for all three numbers.
Image		Will make the field available for an image. Image Constraints: Enable this option if you want to set constraints on the image. Minimum Size of the image (pixels): Specify the minimum allowed image dimensions in pixels. Both width and height must be at least this large. Aspect Ratio (width x height): Specify the aspect ratio of the image. Allowed Error (%): Specify the maximum stretch limit for both with and height of the image. This is in relation to the defined aspect ratio.
Video		Will make the field available for a video.
Font		Will make the field available for a font.
Geometry		Will make the field available for a geometry.
Material		Will make the field available for a material.
Map		Will make the field available to present and edit a map.
Custom		Using the custom type the media- and XSD- type can be freely specified.
List		Lists may be modified by adding and removing columns in the Field Tree . To add columns to a list, right-click the columns node under the list field node in the Field Tree and select Add column . To remove a column, select the column field in the Field Tree and press the Delete button, or right-click it and select Delete field . List fields are fundamentally different from scalar fields. It is therefore not possible to change a list type to a scalar type and vice versa. Minimum number of rows: Defines the minimum allowed number of rows in the list. Maximum number of rows: Defines the maximum allowed number of rows in the list.

 **Note:**
The user must be aware of available control plugins in the template that have been exposed by the scene designer in Viz Artist.

Data Entry

There is a Data entry drop-down list available for all scalar field types. There are three choices in this drop-down list, namely [Manual](#), [Choose from list](#) and [Enable feed browser](#).

Manual does not open up to any additional settings for the field.

Choose from list allows the template designer to present the right content in a drop-down list.

Enable feed browser allows the user to browse for an entry.

For more information see [Data Entry](#).

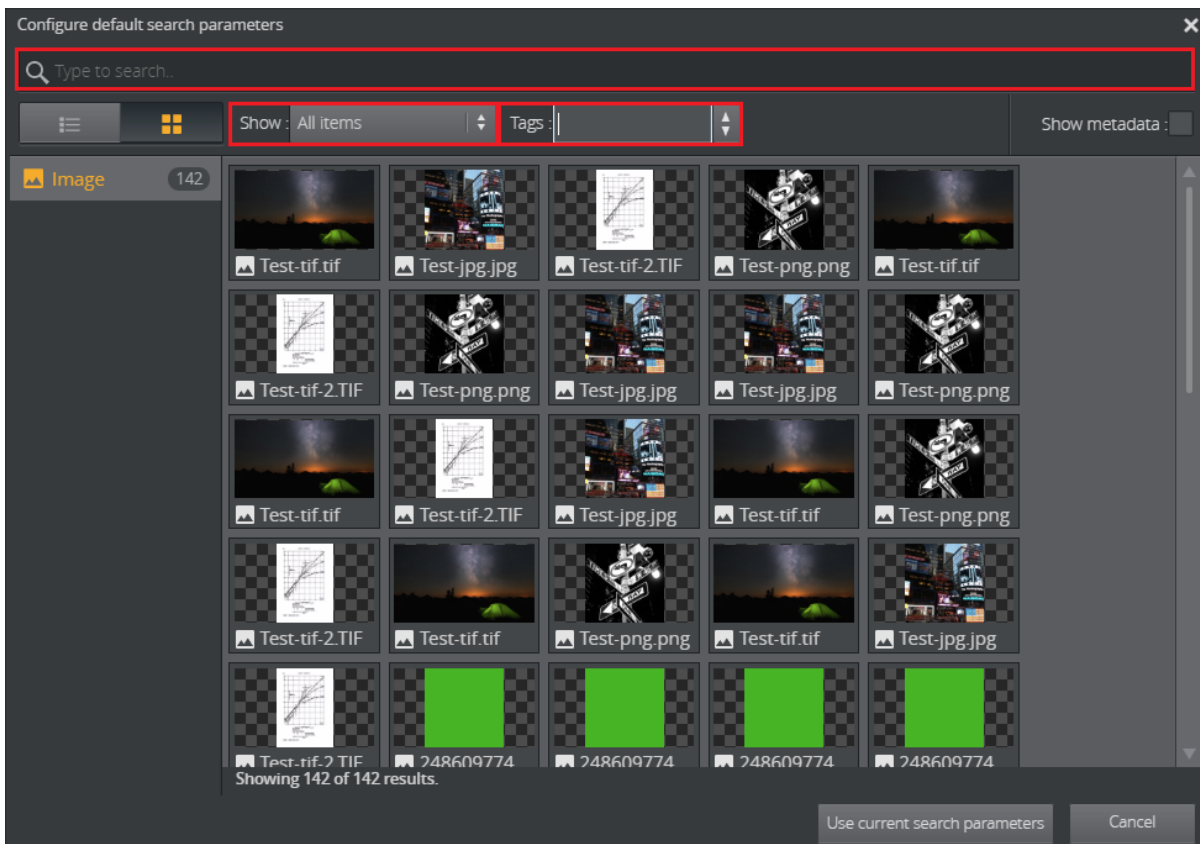
Default search parameters

For the types Image, Video, Font, Geometry, and Material it is possible to define default search parameters that will be used by the media search that is launched when the user clicks on the field.

To define the search parameters for a field, click the **Set** button to open a media search window.

Set the search parameters by writing something in the search field, selecting whether to show all items or to limit by time from the Show drop-down list, and/or selecting a tag from the Tags drop-down.

Save by clicking **Use current search parameters**.



5.2 Settings

The following template settings are available in the **Settings** tab:

- [Duration](#)
- [Track](#)
- [Title Generation](#)

The screenshot shows the 'Settings' tab with the following fields:

- DURATION**
 - Default :
 - Minimum :
 - Maximum :
- TRACK**
 - Name :
 - Index :
- TITLE GENERATION**
 - Format :

5.2.1 Duration

The duration setting can be used to specify the duration of a graphic on a timeline. A default value is used when unspecified. Minimum and maximum duration values may also be specified. If these are set to the same number the item will get a fixed duration.

The close-up shows the 'DURATION' section with the following fields:

- Default :
- Minimum :
- Maximum :

5.2.2 Track

Setting a **Name** in the track settings allows for grouping of graphics in the timeline editor, see [Track settings displayed in the timeline editor](#). The **Index** sets the position of the group in the timeline editor, where 0 is the lowest position.

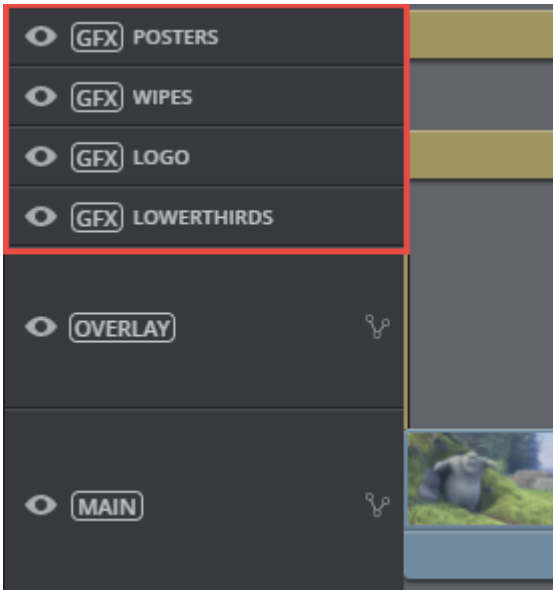
WARNING!
All templates using the same track name must have the same index.

The close-up shows the 'TRACK' section with the following fields:

- Name :
- Index :

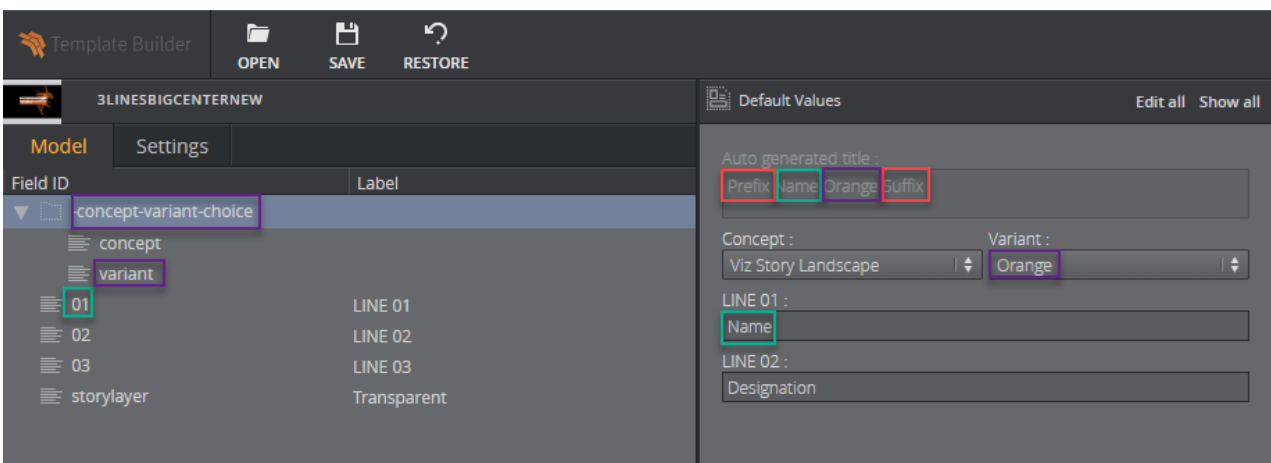
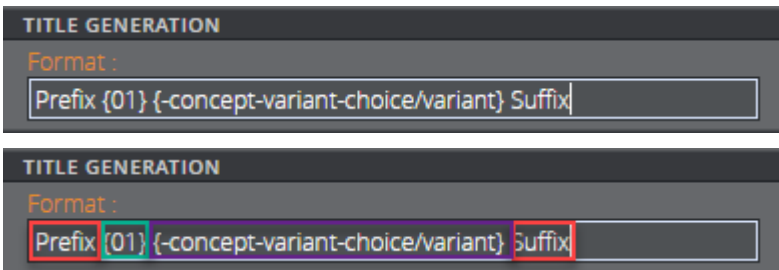
Track settings displayed in the timeline editor

The red rectangle highlights the different tracks in the timeline editor. The position of these tracks is set using the [Track](#) setting.



5.2.3 Title generation

The title generation setting allows for auto-generation of the title. The title can be plain text or it can be a placeholder for one or several field values, or it can be a combination. The placeholder is the {Field ID}. An example combining plain text, field name, and subfield name is shown below:



A template title can be auto-generated by combining one or several of these options:

- Normal text - plain text (red)
- {Field ID} - will be substituted by the value of the field (green)
- {Field ID/subfield ID} - will be substituted by the value of the subfield (purple)
- {listfieldname/#index/cellname} - will be substituted by the value of the field in a row in a list. Note that the index is zero-based.



WARNING!

The auto-generated title does not get trimmed inside Vizrt web clients. However, if the title is longer than 128 characters it will be trimmed when dragging out the MOS XML file due to size constraints. This will affect the element title in the newsroom system.

5.3 Data Entry

The Data Entry field property specifies how the user should fill in the field value. The property can take three values which are explained below:

- [Manual](#)
- [Choose from list](#)
- [Enable feed browser/Parent feed browser](#)

5.3.1 Manual

Selecting **Manual** in the Data entry drop-down list does not give access to any additional settings for the field.

5.3.2 Choose from list

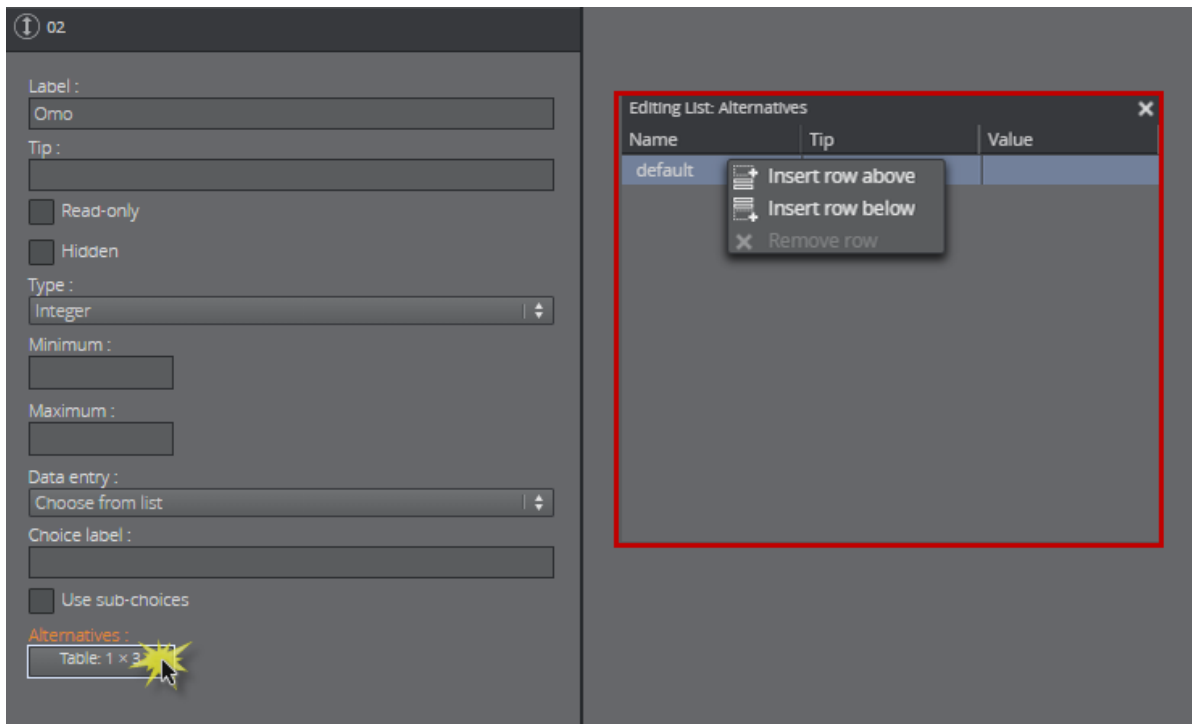
Selecting **Choose from list** allows the template designer to present the content in a drop-down list. This functionality may in some cases make it easier and less error-prone to fill the template with the right content.

One example where the **Choose from list** functionality can be useful is when a Control Object moving (Omo) plugin is accessible in the template. Scenes using Omo plugins will originally be presented as integer values for the different elements in the Fill In Form. The **Choose from list** option can assign text to these values to make it easier to select the right element.

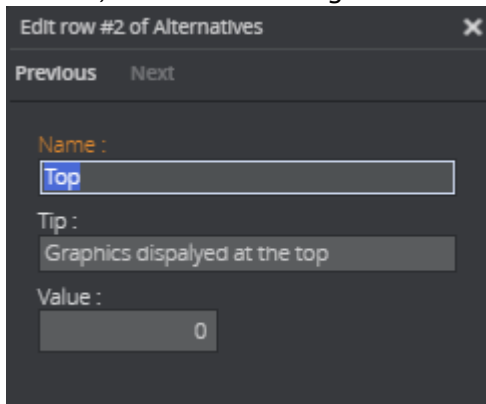
An example of how this can be done is shown below.

In this example, there is a scene that can be displayed at the top, in the middle or at the bottom in the graphics. For the Omo plugin, these positions correspond to the values 0, 1 and 2 respectively. To assign text to these values, do the following:

- Mark the Omo Field ID in the Field Tree.
- Select the **Choose from list** option in the Data entry drop-down list.
- When clicking on the **Alternatives** button, a new window appears. Right-click to design the table by inserting or removing rows. Click in a selected row or press F2 to do inline edits.



- Double-click the table or press Return to insert **Name**, **Tip**, and **Value**. Click **Next**, tab or Ctrl + down, to continue filling the table.



- In the table below the values now correspond to text. Exit the table when it is filled out

Name	Tip	Value
Top	Graphics dispalyed at the top	0
Center	Graphics displayed at the center	1
Bottom	Graphics displayed at the bottom	2

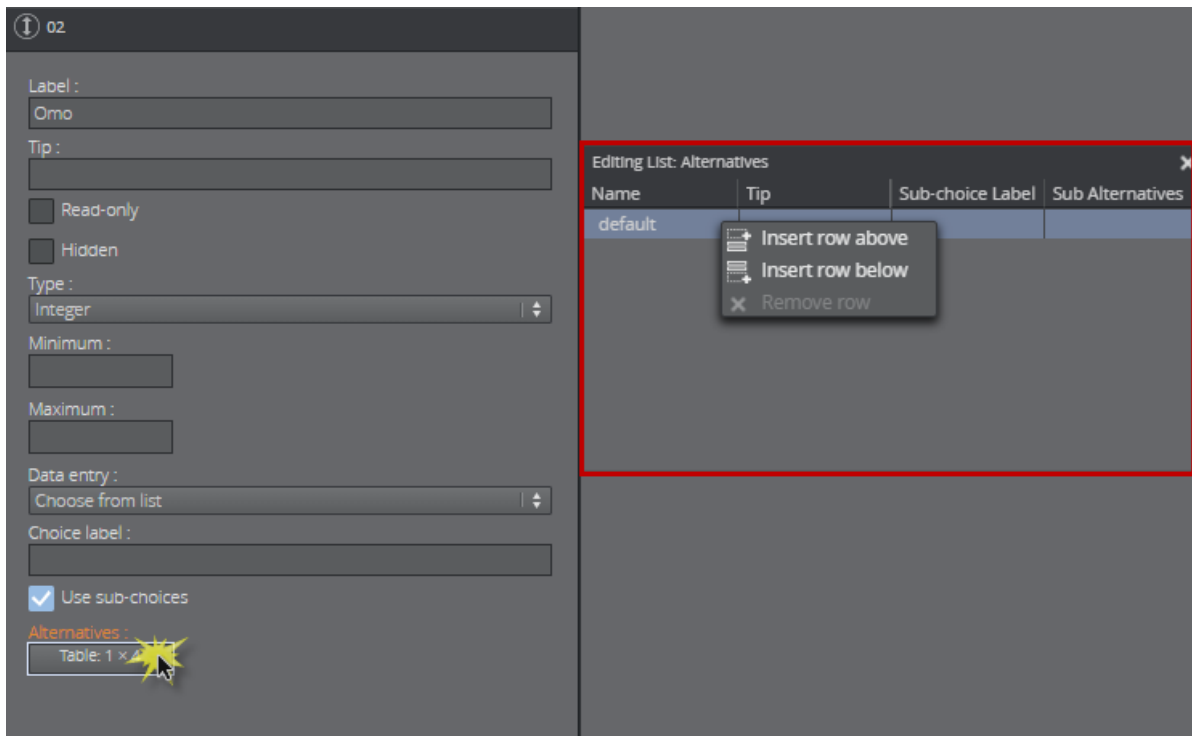
- The Omo field in the Fill In Form now contains a drop-down list containing the alternatives created above as text, as opposed to an integer field where the user would have to remember which integer corresponds to which position.

Use Sub-Choices

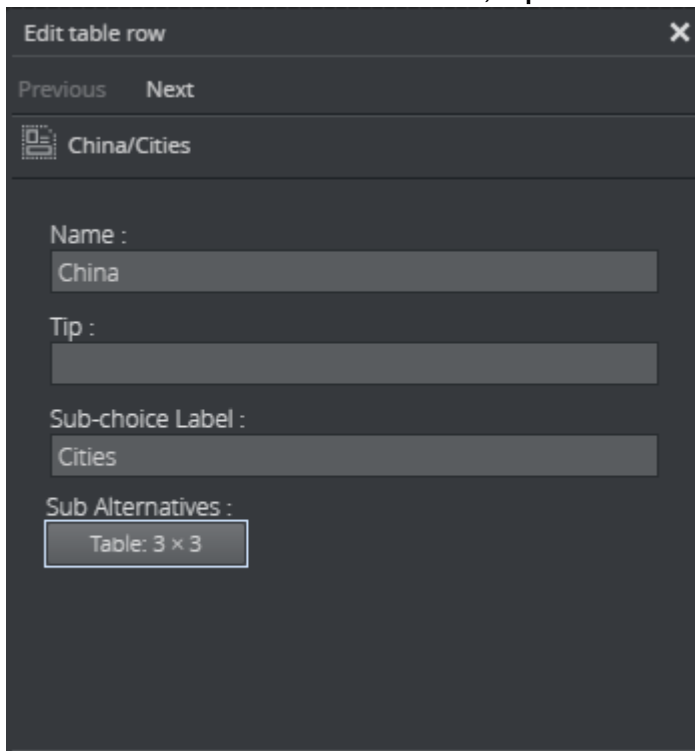
Selecting the **Choose from list** option, a checkbox is made available called **Use sub-choices**. This functionality allows the set-up of several sub-choices for each choice.

For example, if the choices list different countries, sub-choices could list cities in each of the countries. See example below:

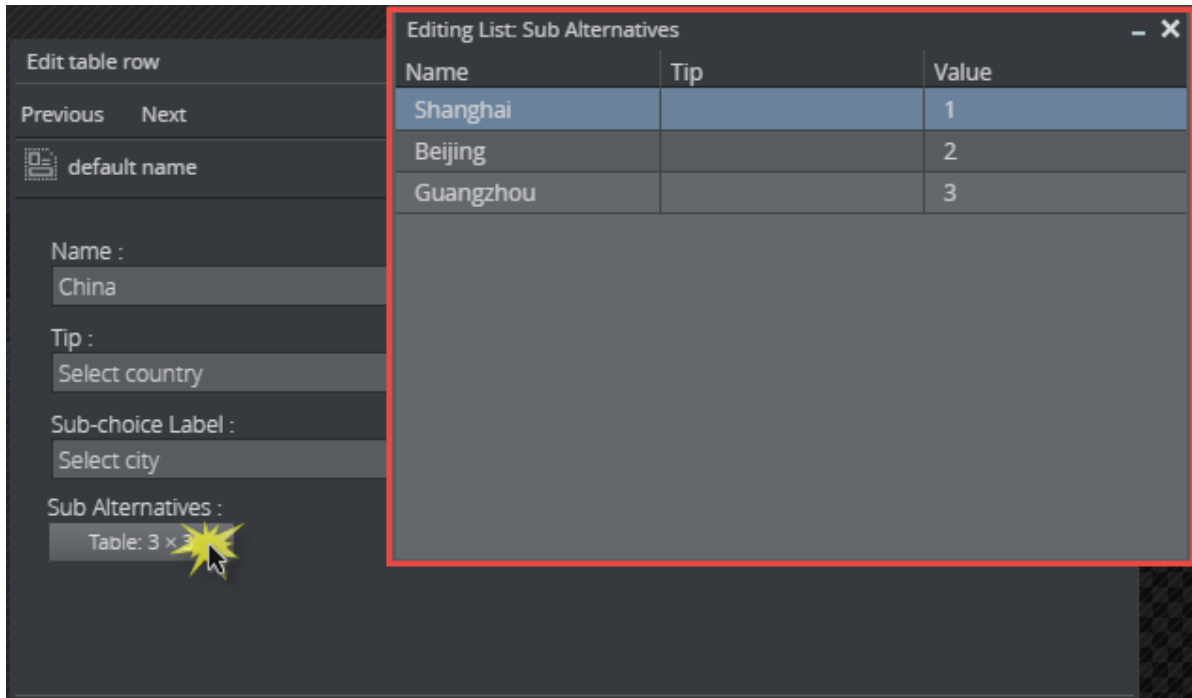
- Mark the desired Field ID in the Field Tree.
- Select **Choose from list** and tick the **Use sub-choices** check box.
- When clicking on the **Alternatives** button, a new window appears. Right-click to insert or remove rows.



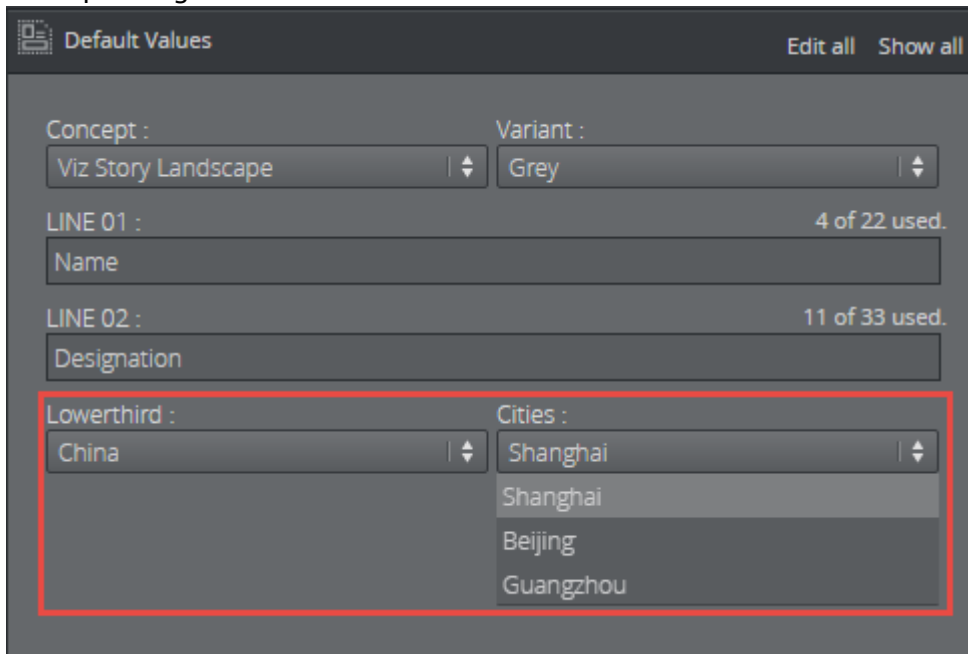
- Double-click the table to insert **Name**, **Tip** and **Sub-choice Label** which in this case is *Cities*.



- Click the **Sub Alternatives** button to add the sub-alternatives.



- Exit the tables when they are filled out.
- Instead of a text field in the Fill In Form, the field now contains two drop-down lists; the main choices, which in this case is a list of countries, and sub-choices with corresponding cities.



5.3.3 Enable feed browser/Parent feed browser

This option specifies that the field should get its value from some property of an atom feed entry. If the field is a subfield of another field that has enabled feed browser, the option will be named *Parent feed browser*. Otherwise, it will be named *Enable feed browser*.

If the Enable feed browser option is selected, a **Browse** button will appear next to the field in the fill-in form. Clicking this button will open the **Feed Browser** dialog. In the Feed Browser, the atom entries of the feed will be presented (with thumbnails if available), and one of the entries can be selected. Information from the selected atom entry will be used to fill in the feed browser enabling field and its subfields.



Note:

In order to be able to fill in multiple fields from a single selection in the feed browser, those fields must be subfields of the field enabling the feed browser.

Atom feed URL

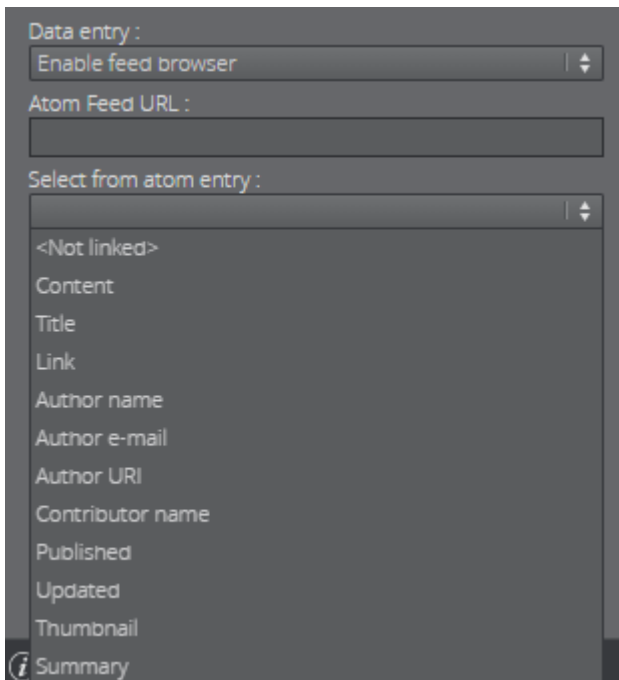
This field property is available only for feed browser enabling fields (not for fields using *parent feed browser*). The property specifies the URL of the atom feed that should be used to fill in the value of this field and a selection of its subfields.

The image shows a configuration dialog for a field. It has three main sections:

- Data entry:** A dropdown menu currently showing 'Enable feed browser'.
- Atom Feed URL:** A text input field, which is highlighted with a red rectangular border.
- Select from atom entry:** A dropdown menu that is currently empty.

Select from atom entry

This field property specifies which information in the atom entry selected in the Feed Browser that should be used to fill in this field. The following options are available, but the possible options for a given field depend on the type of the field (the atom namespace prefix represents the *http://www.w3.org/2005/Atom* namespace, and the media namespace represents the *http://search.yahoo.com/mrss/* namespace):



- **<Not linked>** - The field is not linked to the atom feed, and must be filled in manually.
- **Content** - The field is linked to the content of the `atom:content` element in the atom entry.
- **Title** - The field is linked to the content of the `atom:title` element in the atom entry.
- **Link** - The field is linked to the `href` attribute of the `atom:link` element in the atom entry. Which link entry to pick depends on the *Link-rel in atom entry* property and the type of the field. (The first link with a correct `rel` attribute and a type that matches the type of the field will be chosen.)
- **Entry** - The field is linked to atom entry itself.
- **Author name** - The field is linked to the content of the `atom:name` element inside the relevant `atom:author` element. (If the entry itself contains an `atom:author` element, that will be used. Otherwise the `atom:author` element of the feed itself will be used.)
- **Author e-mail** - The field is linked to the content of the `atom:email` element inside the relevant `atom:author` element. (If the entry itself contains an `atom:author` element, that will be used. Otherwise the `atom:author` element of the feed itself will be used.)
- **Author URI** - The field is linked to the content of the `atom:uri` element inside the relevant `atom:author` element. (If the entry itself contains an `atom:author` element, that will be used. Otherwise the `atom:author` element of the feed itself will be used.)
- **Contributor name** - The field is linked to the content of the `atom:name` element inside the `atom:contributor` element in the atom entry.
- **Published** - The field is linked to the content of the `atom:published` element in the atom entry.
- **Updated** - The field is linked to the content of the `atom:updated` element in the atom entry.
- **Thumbnail** - The field is linked to the `url` attribute of the `media:thumbnail` element in the atom entry.
- **Summary** - The field is linked to the content of the `atom:summary` element in the atom entry.

Note:
Unless you hide the linked field or make it read-only, it may also be filled in manually.

Link-rel in atom entry

This property is only available if *Link* is selected in the *Select from atom entry* property. It specifies what the `rel` attribute of the link element in the atom entry should be.

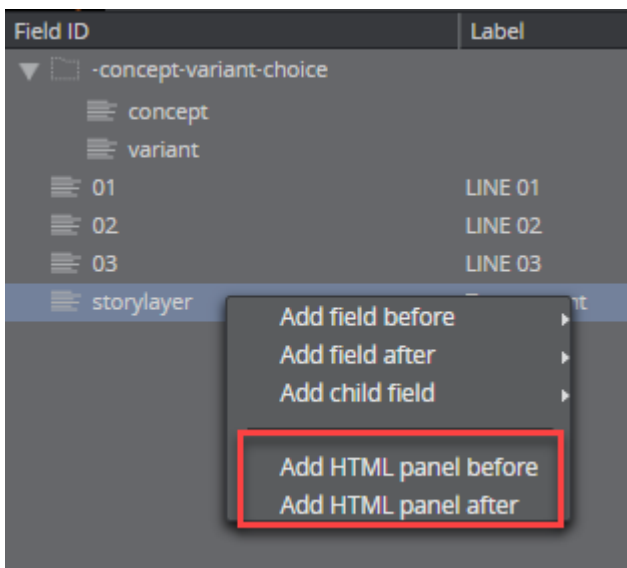
5.4 The HTML Panel

As part of the template customization workflow, the user can add an HTML panel to the template, which is a functionality that gives the user full control over the template by using custom scripting and logic to build the template. The template can host a webpage by entering a web address in the Template Builder. Normally the aim is not to host a single webpage, but rather make a data integration HTML template.

Some examples on how to use HTML templates are shown in the [Creating HTML templates](#) page.

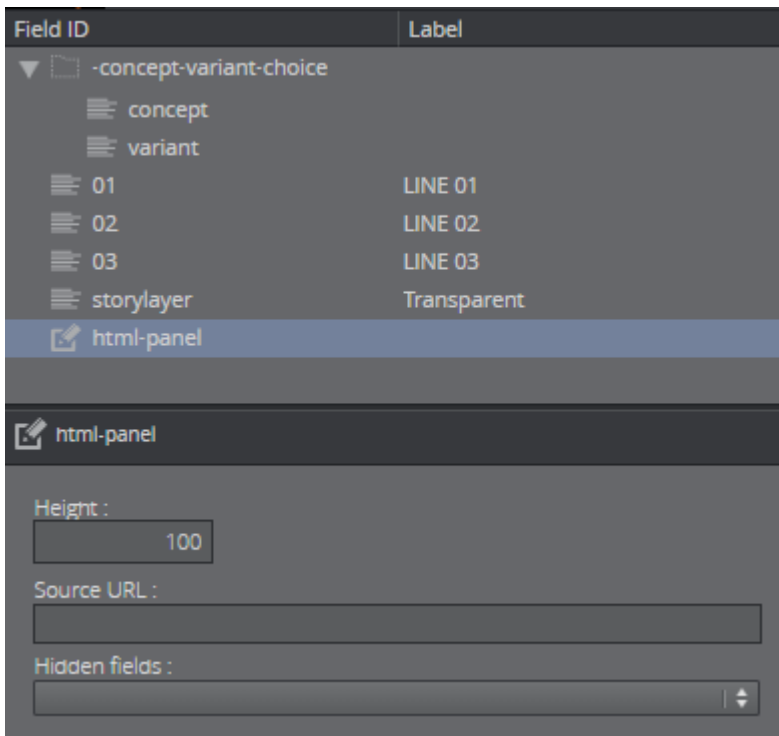
Add an HTML panel

The option of adding an HTML panel is available by right-clicking in the field ID list in the field tree and select **Add HTML panel before/after**.



Enter the ID of the of the new panel in the dialog that appears.

Selecting the new HTML panel field ID in the field tree, its properties become available in the [Field Properties](#) window.



Adjust the size of the HTML panel shown in the Fill In Form using the **Height** field.

In the **Source URL** type in the web address.

The **Hidden fields** drop-down list allows the user to hide available fields in the Fill In Form.

5.4.1 Browser Caching

The user might experience browser caching behavior when trying to update and display changes in the custom HTML template in Template Builder. This is standard browser behavior and Template Builder does not have control over caching resources that the HTML file itself includes.

To prevent this browser behavior, following actions can be made:

1. Make sure the URL's to the resources are unique upon reload.
2. The web server serving the resources could be configured to send Expiry headers set to 0. More detailed information on how caching works and how to modify these is out of scope for this documentation
3. Disable caching on the browser side. This is done differently per browser.

5.4.2 Creating HTML templates

A few examples on how to create HTML templates will be shown below:

- [Set up a simple custom HTML template](#)
- [Connect the custom HTML template to a Viz Pilot template - the simple way](#)
- [Connect the custom HTML template to a Viz Pilot template - the advanced way](#)
- [Redesign Concept/Variant fields](#)

To fully understand the workflow in these examples some basic knowledge about web technology (javascript, HTML, CSS) is required. A detailed description of the content in the files used will not be given. However, in addition to this documentation, there is also an API documentation bundled with the product describing the API a web developer creating custom HTML templates has access to. This can be found at <http://<pilotdataserverhost>:8177/app/templatebuilder/js/doc/PayloadHosting/index.html>.

Keep in mind that the following examples are a proof of concept and show a limited possibility of what can be done using the customized workflow. Using it in more advanced ways opens up to full user control of the template using custom scripting and logic where the user has 100% control over what to do. In general:

- You can use simple HTML `<input>` or `<textarea>` fields that contain an `id="field_<fieldid>"` which will automatically have a bi-directional connection
- You can take control of the function mapping, and use javascript to basically do whatever you want.

Set up a simple custom HTML template

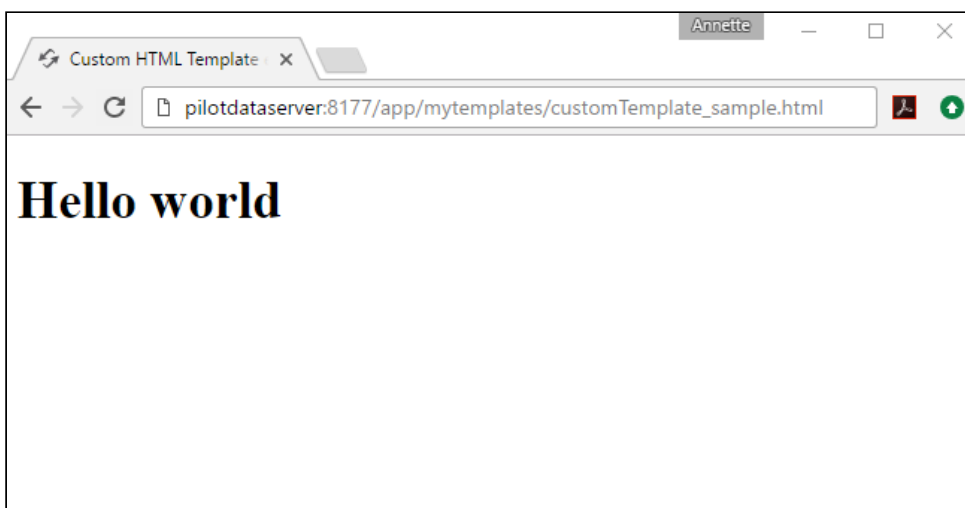
The following example will show how to set up and show a simple custom HTML template in Template Builder.

We will work with a template that will show the message **Hello world** when opened in a browser, see the picture below. Three necessary files, including the HTML template, must be located in the same folder (C:\Program Files\Vizrt\Pilot Data Server\app\mytemplates\).

The files are:

1. `customTemplate_sample.html` - The custom HTML template
2. `customTemplate_sample.js` - The javascript part for the template
3. `payloadhosting.js` - Needs to be present as it connects everything

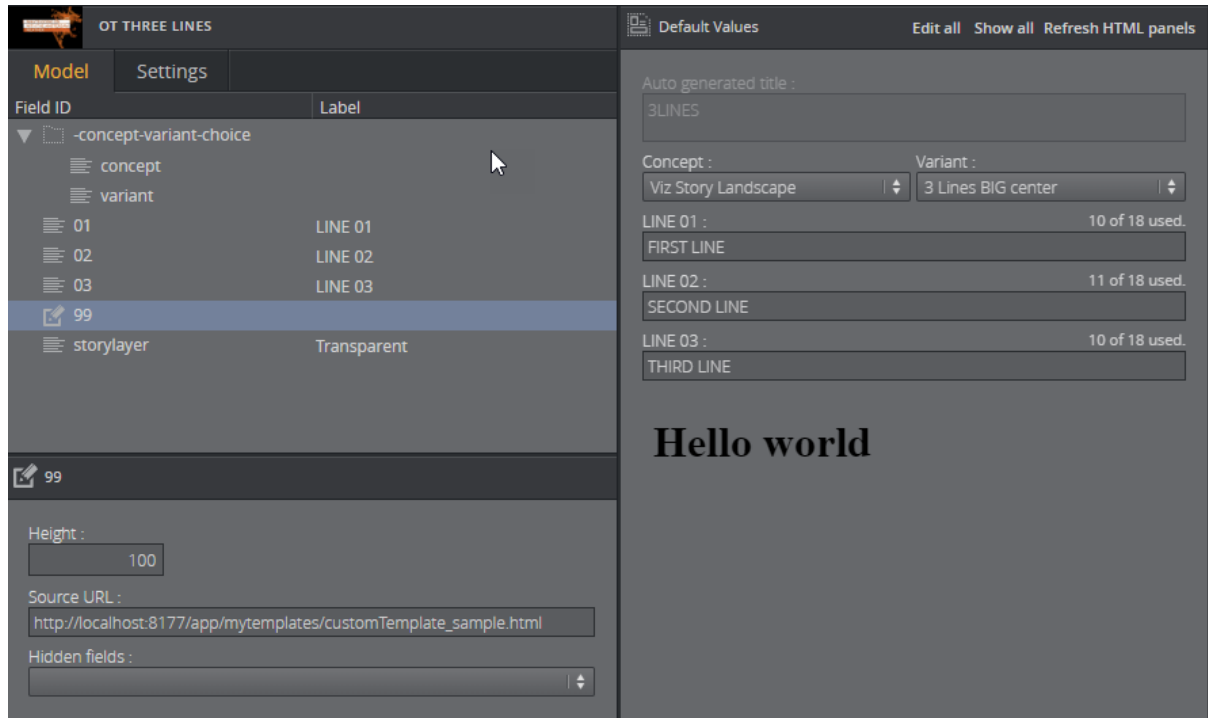
The URL in the image below points to the location of the .html file mentioned above.



How to see the custom HTML template inside Template Builder

In the image above the template is shown in a browser, but now we want to see the template inside Template Builder.

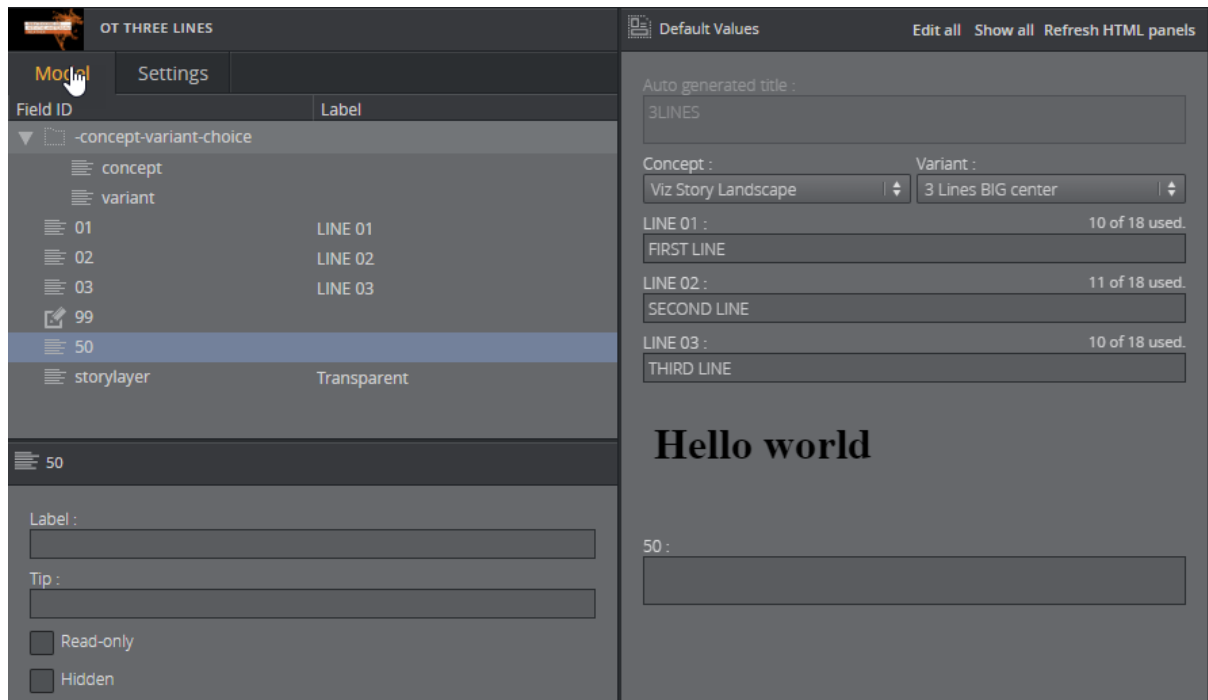
- Open a template and add an HTML panel as explained [here](#).
- In the URL field enter the URL of the custom HTML template, in this case, the URL from the picture above.
- Now the message **Hello world** appears in the Fill In Form:



Connect the custom HTML template to a Viz Pilot template – the simple way

Following the example above, we can establish a two-way communication, or *bind fields*, between the HTML template and the opened pilot template. This is done by adding a new field to the template in the following way:

- Right-click in your HTML panel field, choose **Add field before/after** and select **Plain text**
- Give it the ID **50**
- A new field appears:



Then the `<body>` block in custom HTML template (Template_sample.html) that previously contained:

```
<body>
  <h1>Hello world</h1>
</body>
```

must be replaced with:

```
<body onload="vizrt.payloadhosting.initialize()">
  <label for="field_50">My input field:</label>
  <input name="field_50" type="text" id="field_50">
</body>
```

Saving the HTML file and clicking **Refresh HTML panels**, reloads the custom HTML template with the changes just made. Now a bi-directional connection between the custom template and pilot template have been established. When typing either inside the template or inside the field with ID 50, both fields should be updated at the same time. This way of binding fields will work for any HTML fields that have value support, which is typically `<input>` types and `<textarea>`.

The screenshot shows the Template Builder interface. On the left, the 'Model' tab is active, displaying a tree view of the model structure. The 'Field ID' and 'Label' columns are visible. The '99' field is selected, showing its settings: Height (100), Source URL (http://localhost:8177/app/mytemplates/customTemplate_sample.html), and Hidden fields. On the right, the 'Default Values' tab is active, showing the auto-generated title '3LINES' and the concept 'Viz Story Landscape' and variant '3 Lines BIG center'. Below this, the HTML output is displayed, showing three lines of text: 'FIRST LINE', 'SECOND LINE', and 'THIRD LINE'. A text input field is shown with the value 'The custom template is updated'.

The javascript file automatically tries to find input elements in the HTML that match the ID of fields inside the template. Adding the `id="field_50"` to the `<input>` element inside the HTML template is all that is needed for the two-way communication to be set up since we added a field with the ID of 50. Any amount of these binding fields can be established in the exact same way; the way they are mapped are via ID.

This gives a simple way of setting up a binding field. Note that updating the `<input>` elements programmatically will still send data back and forth, which is great for automated data integration such as fetching live sports data.

Note

To prevent two editors for the same field being visible at the same time, use the **Hidden fields** setting inside the HTML panel settings.

Connect the custom HTML template to a Viz Pilot template – the advanced way

This example will go more into detail than [the simple example above](#), and use a bit more scripting to give 100% control over the template. The three files mentioned in the [Setup a simple custom HTML template](#) example, are also used in this example.

A list of functions will be created where we can bind fields. By adding the following above the `document.ready()` function in the `customTemplate_sample.js` file:

```
// Will be called when the field with id "01" changes
function on50Changed(value) {

}
```

and the following inside the `document.ready()` function

```
var pl = vizrt.payloadhosting;
pl.initialize();
pl.setFieldValueCallbacks({ "50": on50Changed });
```

we are setting up a way for a custom javascript function to be called upon detecting a change. When `field_50` receives a change from the host, the function will be called with its new value as a parameter.

In the following, some changes will be made to the HTML file to demonstrate that we can use custom HTML/javascript to do something with these values.

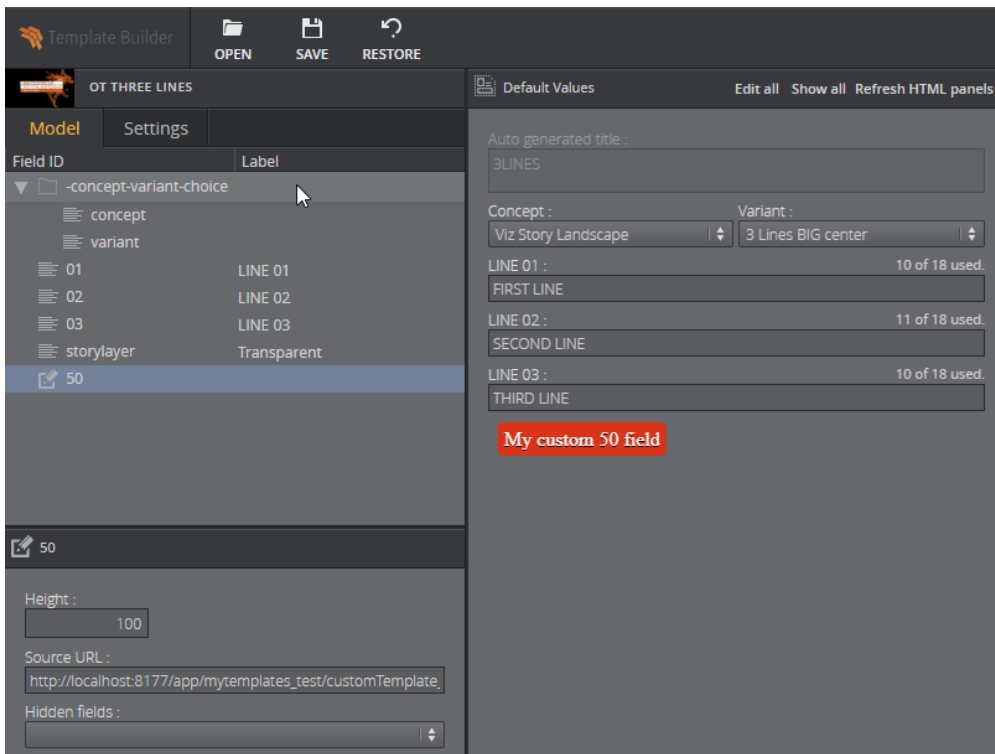
Inside the HTML file, the entire body is replaced with:

```
<body>
  <span id="myfield" class="sample red">My custom 50 field</span>
</body>
```

To accompany some CSS to style the text, add the style tag after the closing `</head>` tag and before the `<body>` tag:

```
<style>
  .sample {
    padding:5px;
    color:white;
    border-radius:5px;
    text-shadow:0 1px 0 black;
    background:red;
  }
  .green {
    background:green;
  }
</style>
```

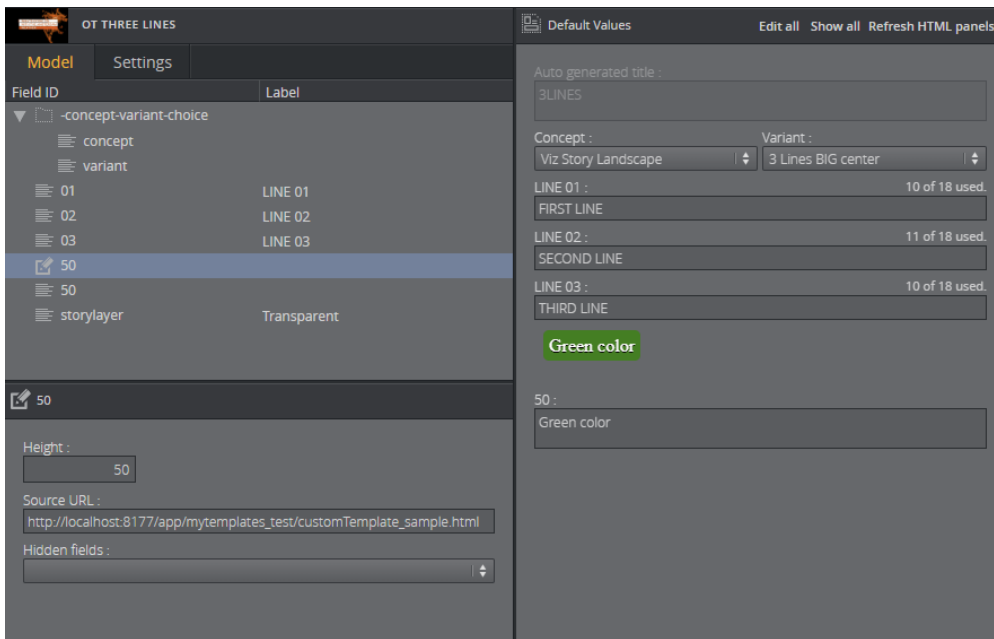
This gives the following output in Template Builder:



Adding a bit more custom logic, we will make the background color green when there is a text value that is longer than 5, or shorter than 20 characters. The function is expanded with the following function:

```
function on50Changed(value) {
  var myField = $("#myfield");
  myField.text(value);
  if (value.length > 5 && value.length < 20) {
    myField.addClass("green");
  } else {
    myField.removeClass("green");
  }
}
```

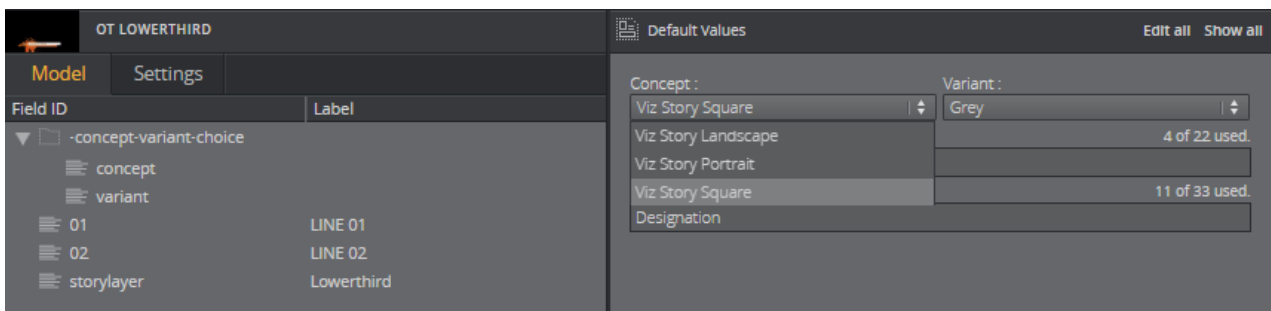
Refreshing the HTML panel the background color should change to green dynamically when typing.



Redesign Concept/Variant fields

The next example shows how to present the concepts and variants in a template in a different way. A custom HTML template, javascript, and a bootstrap theme (for styling) are used to achieve this, but will not be shown in this example.

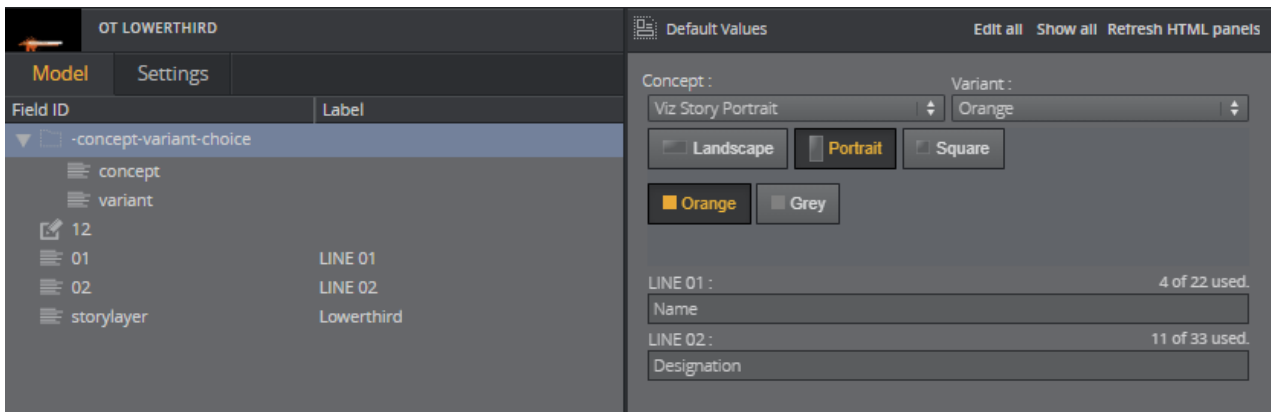
We have the following concepts, **Viz Story Landscape**, **Viz Story Portrait**, **Viz Story Square**, and variants, **Grey**, **Orange**, originally available as drop-down lists in the Fill In Form:



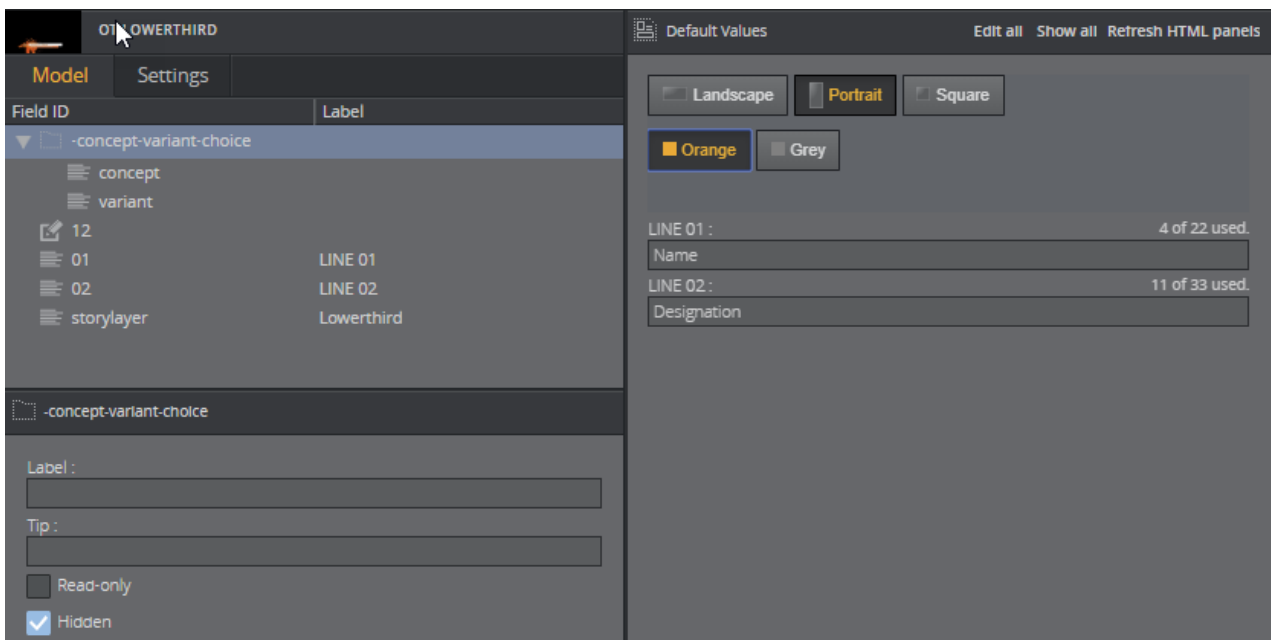
This field has more than one value since it has an ID, **-concept-variant-choice**, and two values beneath it, **concept** and **variant**. You can access a value using slash "/" to navigate in the list, for example accessing concept use the ID, **-concept-variant-choice/concept**.

The new presentation of the concept and variant values are in form of buttons.

This example has a two-way binding support for both concept and variant which means that clicking on the new buttons will update the original drop-downs and vice versa:



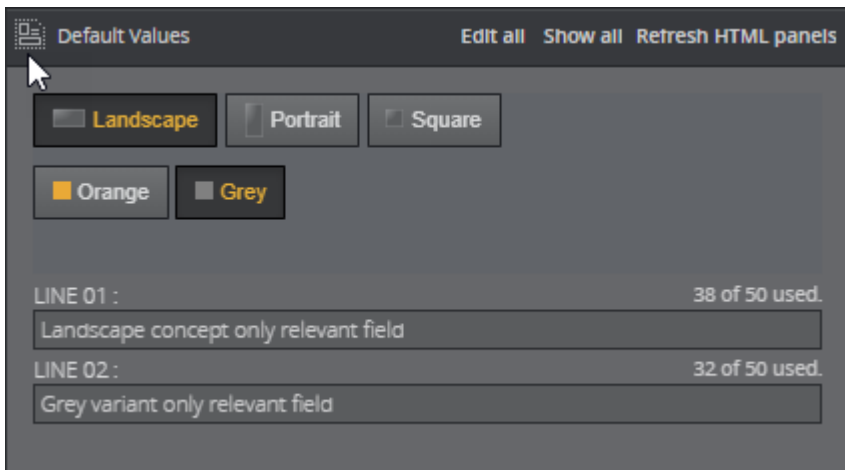
Since we have created a nicer way to select concept and variant, the drop-down lists are no longer needed and can be set as **Hidden** in the properties window.



Controlling the auto-generated Fill In Form from the HTML template

It is possible to dynamically set visibility and read-only attributes which opens up the possibility to filter the auto-generated form based on the choices made in the custom HTML template.

In this example, the LINE 01 text field should only be visible when the Landscape concept is active.



In the javascript used in this example, there is a function called `onConceptChange(value)` which is called when the concept changes from the host. Adding the following code block inside the `onConceptChange` method block will check if the Viz Story Landscape concept is chosen. If it is, it will display the field with ID 01 in the Fill In Form:

```
if (value == "Viz Story Landscape") {
  vizrt.payloadhosting.setFieldVisibility("01", true);
} else {
  vizrt.payloadhosting.setFieldVisibility("01", false);
}
```

This will only work if the value has been changed by the auto-generated concept/variant drop-down lists. To make it work from using the custom buttons too, the same logic is needed there. Replace the following code block:

```
$("#concept .btn").on("click", function() {
  pl.setFieldText("-concept-variant-choice/concept", $(this).data("concept"));
  clearActiveConcept();
  $(this).addClass("active");
});
```

with

```
$("#concept .btn").on("click", function() {
  var conceptValue = $(this).data("concept");
  vizrt.payloadhosting.setFieldVisibility("01", conceptValue == "Viz Story
Landscape");
  pl.setFieldText("-concept-variant-choice/concept", conceptValue);
  clearActiveConcept();
  $(this).addClass("active");
});
```


Now, clicking on Viz Story Square or Portrait concepts the text field with ID 01 will disappear, but it will show if the Viz Story Landscape concept is selected. This is a powerful feature as you can customize available editing options based on certain conditions set in the template.

6 The Fill In Form

The window in the middle of the Template Builder application is the auto-generated Fill In Form for the graphics template. It displays the default values in the template which are editable fields where the user can add content. The types of content allowed in the fields are controlled in the [Template Builder Window](#). Fields can be restricted to contain e.g. only text with a certain amount of characters, numbers within a specific range, media placeholders for media assets, or it can be a set of choices in a drop-down list.

As long as the fields in the Fill In Form are exposed controls made by the template designer, the added content will be shown in the [Preview Window](#).

At the top of the Fill In Form there are three options **Edit all**, **Show all** and **Refresh HTML panels**.

Edit all is an option to edit all fields in the Fill In Form. This is to provide a way to edit default values of fields that are set to read-only.

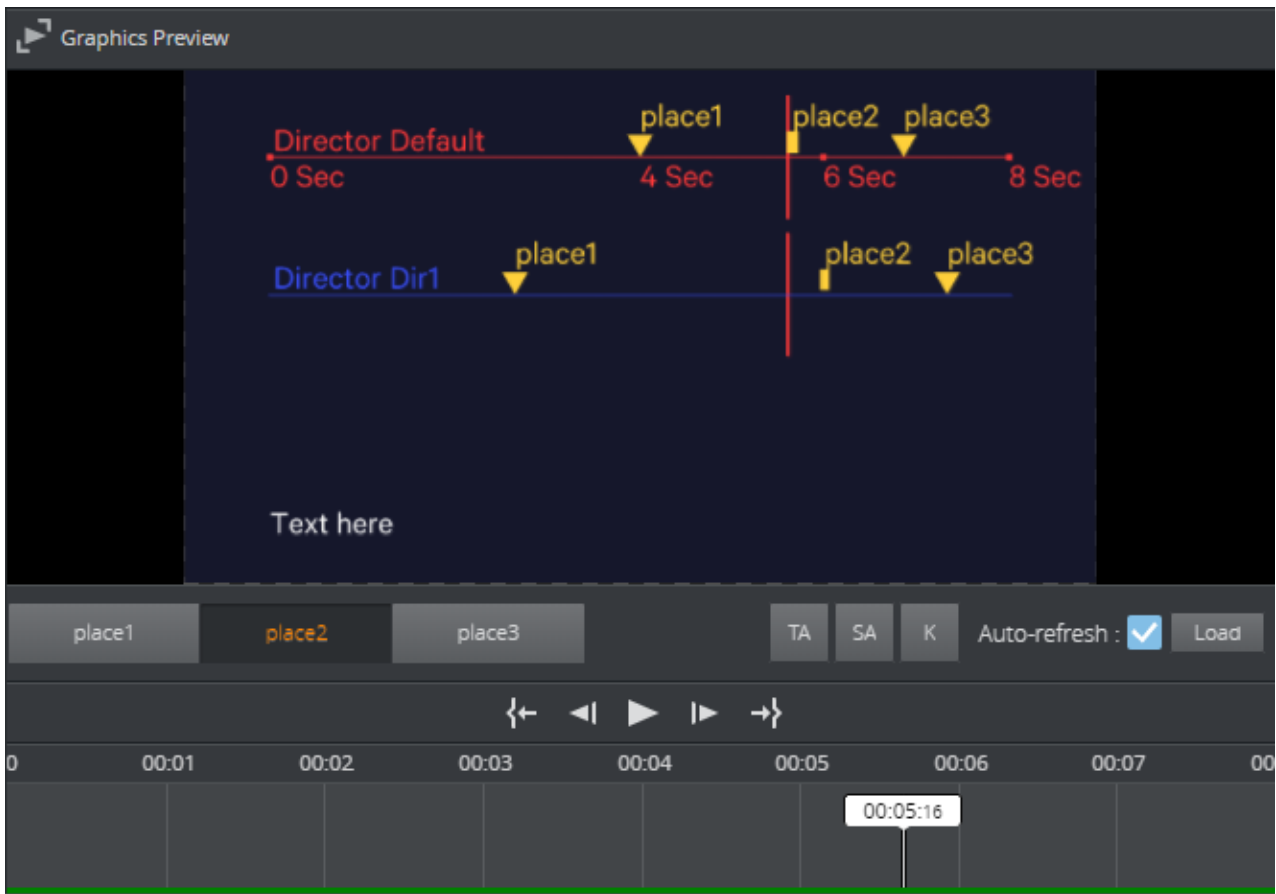
Selecting **Show all** displays all fields in the Fill In Form, even if the fields are set in hidden mode.

Selecting **Refresh HTML panels** will reload the HTML panel(s) in the Fill In Form.

7 The Graphics Preview Window

The **Graphics Preview** window is located to the right in the Template Builder. It displays snapshots of the final output in an ongoing preview process, and provides you with an idea about how the graphics will look when played out in high resolution on a Viz Engine.

The Template Builder sends requests to the Preview Server which manages Viz Engines that provides the snapshots.



The toolbar at the bottom of the Preview Window allows you to adjust how to view the preview of the element graphics:

- **Preview points:** If the scene contains named preview points, such as stop points and/or tags in the Default director, these are shown as buttons on the toolbar. If there is not enough space for the buttons, they appear in a drop-down list instead. Pressing the buttons or selecting an entry from the drop-down list shows a preview of the scene at the given preview point, and the playhead on the timeline jumps to the point in time where the preview point is set.
- **TA:** Show/hide the Title Area.
- **SA:** Show/hide the Safe Area.
- **K:** Show the key signal for the graphics.
- **Load:** Clicking this button will load the animation of the graphics. Once loaded, indicated by a green line at the bottom of the timeline editor, media controls appear allowing for controlling the graphics animation in the Preview Window.

- **Scrub** the timeline back and forth by clicking on it or moving the playhead. If the scene does not have a director called Default, or the Default director does not have a duration the timeline will be disabled. [Setting the minimum duration](#) in Template Builder will enable the timeline. Setting the default duration in Template Builder will change the duration of the timeline if it is enabled.

Info:

Clicking on a preview point to request a preview sends a snapshot request with a named position to the Preview Server. Clicking on the timeline sends a snapshot request with an absolute position to the Preview Server. For further details, refer to the documentation for the Preview Server REST API.

7.1 Auto-Refresh

Auto-refresh is located at the bottom of the Preview Window and is on by default. This means that the Preview Window updates immediately on any changes in the Properties Editor.


Click the checkbox to turn off **Auto-refresh**. To refresh the Preview Window in this mode, click the

Refresh button  Auto-refresh : or click inside the Preview Window.

8 Keyboard Shortcuts

This page lists the available keyboard shortcuts in Template Builder.

Shortcut	Description
Ctrl + O	Open the Open Template dialog where you can select a template to open.
Ctrl + S	Save a template.

 **Warning!**
In Firefox version 65.0.1 and higher, the shortcut **Ctrl + O** will not work as expected.

8.1 Graphics Preview

The player in the **Graphics Preview** window which plays the animation of the graphic has the following keyboard shortcuts:

Shortcut	Description
Space or Ctrl + Space	Play/pause
Shift + I	Go to the in-point.
Shift + O	Go to the out-point.
, (comma)	Move 1 frame back.
. (period)	Move 1 frame forward.