Parameterizationof Graphic Objects

Parameterization allows transferring information from external sources to a graphic object within a composition, during its playout. Depending on a data source, parameterization is divided into two types: External and RSS.
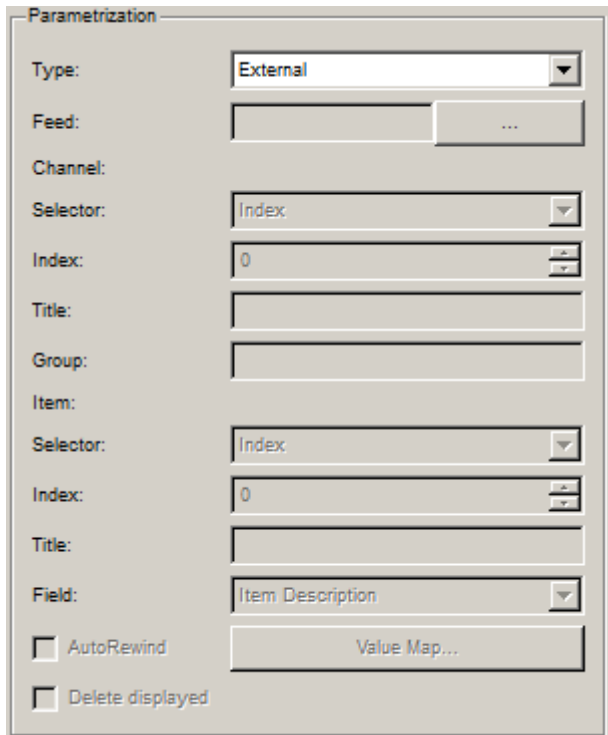
External Parametrization

In this mode, data is loaded from a program channel playlist, text object or action. Preliminarily enter data to the source manually, or configure action generation. This type of parameterization is useful for creating a program in broadcasting studios. It allows, for instance, easy creating captions with a guest name based on one graphic composition.

Parametrized Objects

The following objects have the external type of parameterization:

- **PIP frame** – parameterization used for setting the displayed clip; used for output of weather icons, traffic lights etc.
- **Text object** – using parameterization to load the displayed text. This type is used for the output of text information that does not require being updated during playout: "Now Playing" and "Coming Up Next" captions, age markers, etc.
- **Text counter** – parameterization for initialization of the counter original value; used in "you can skip this ad in" captions etc.

Activation and Configuring

Selecting the External type of parameterization activates the loader of data from an external source.
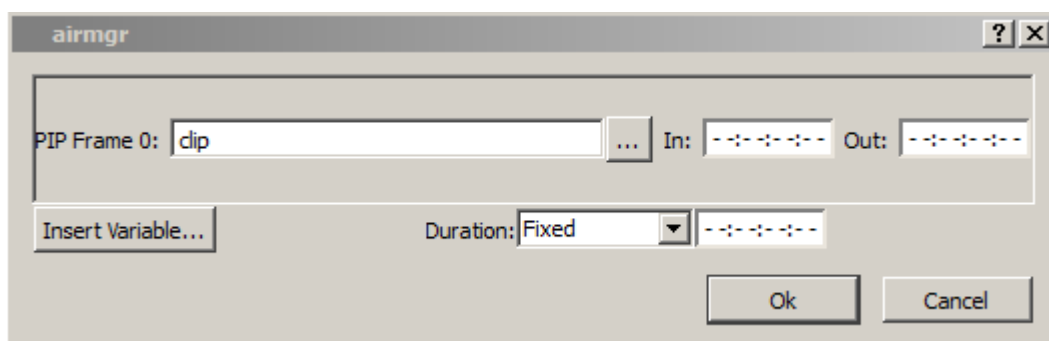
Data Transfer from Playlist
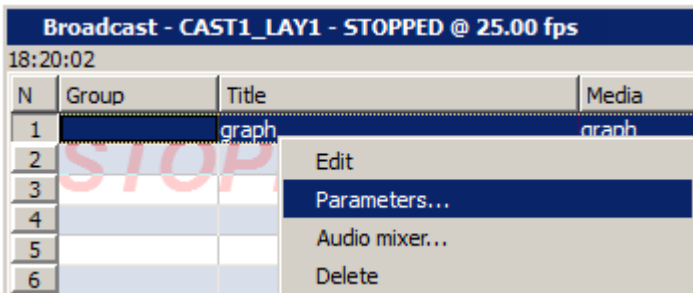
Right-click a graphic composition in the playlist and select the Parameters... menu item. If a composition has External parameterized objects, you will see the object name as in the graphics editor, and its settings.

Parameterized objects receive the specified value or calculated value of the variable.
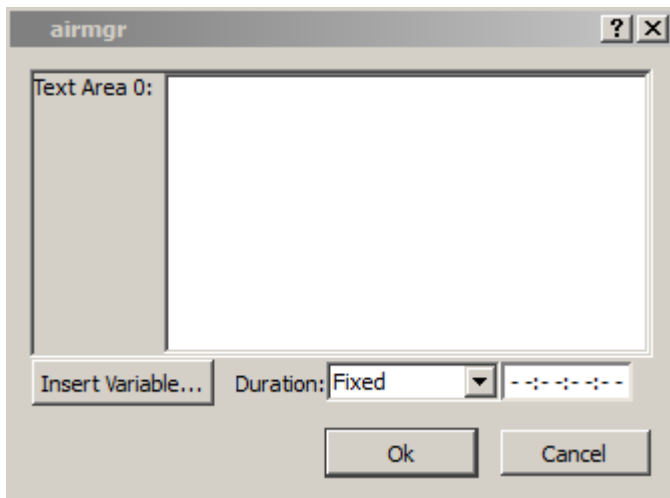
PIP frame

Sending a clip name in the media database and In and Out points for clipping the inserted clip. Based
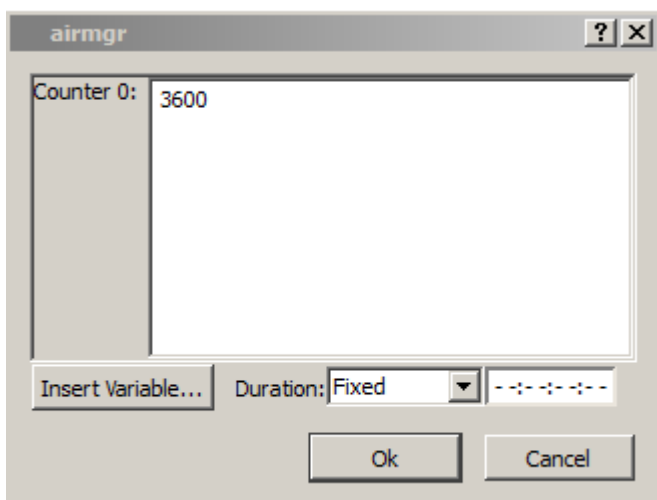
on data in this field, the Duration parameter value can be either calculated automatically or entered manually. In the version 2.4.30, setting In and Out points has been fixed.

Text object

Transferring text. Based on data in this field, the Duration parameter value can be either calculated automatically or entered manually.

Text counter

Sending a value in seconds, depending on a counter type:

- **Wall clock** – shift relative to the current time.
- **Counter** – original counter value.
- **External** – original value.
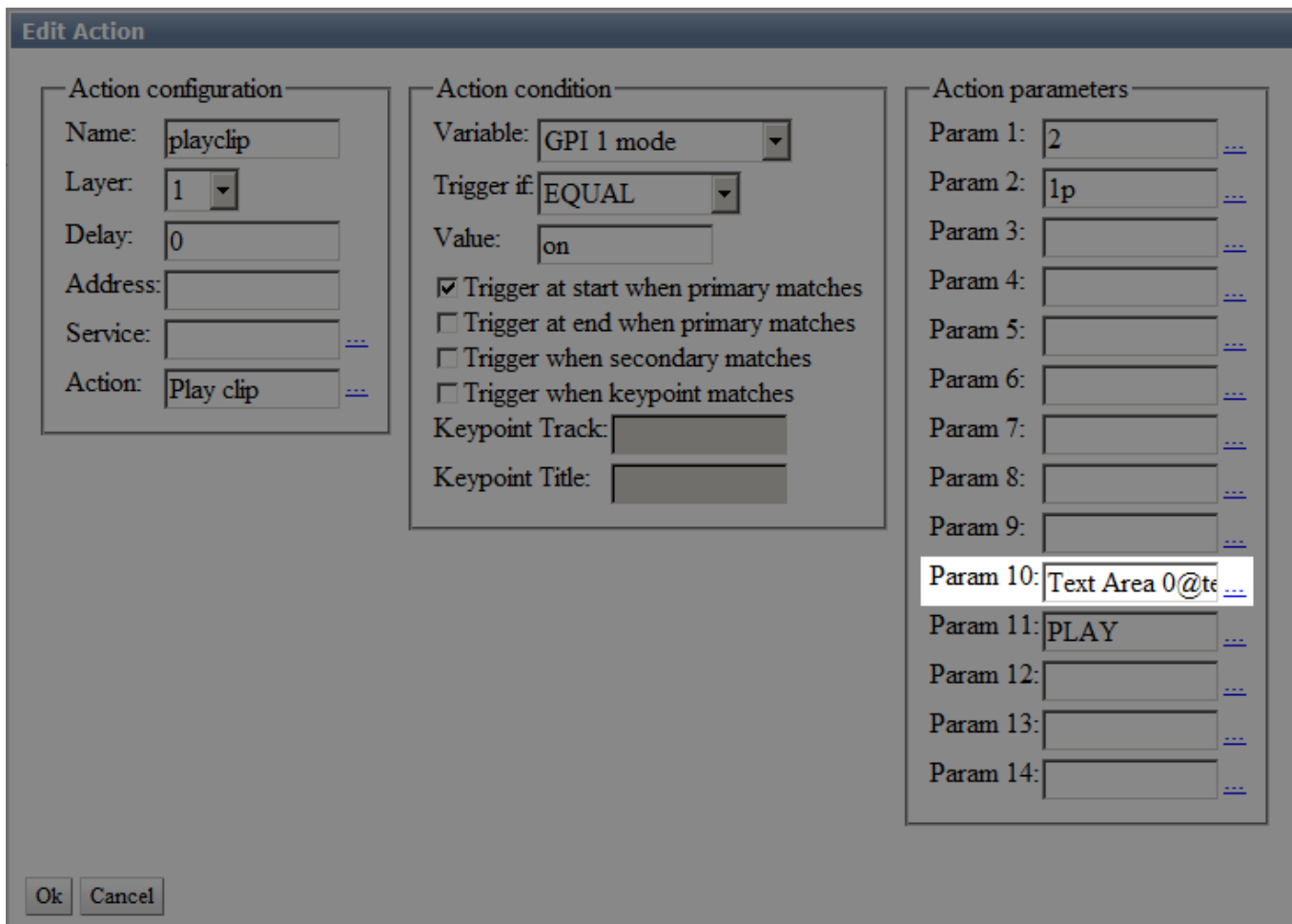- **Date** – shift relative to the current date.

The Duration parameter can be specified manually for this field.

Data Transfer from Text Object

🔧 **Fix Me!**

Data Transfer from Action

Some Action can send parameters to graphic compositions. For example, the Play Clip action in the Param 10 field (Clip Parameters) accepts data for parameterization of the played clip.

The format of data sent for parameterization is the following: `ObjectName@ObjectData[@ObjectName2@ObjectData2][…]`, where:

- `ObjectName` – name of the graphic object within a composition, set in the Name field.
- `@` Symbol – both a separator inside the "object+value" pair, and separator for such pairs.
- `ObjectData` – data sent to the graphic object. Divided into:
  - **text** – the value is specified in the ObjectData field when saving the action.
  - **simple variable** – contains a value read off a playlist column. For example, the `$(title)` variable contains the played clip title. To add a variable, use the variable wizard launched by pressing "…".
  - **calculated variable** – the value will be calculated at the moment of Action launch: for example, the time left until the next clip starts. To add a calculated variable, use the variable wizard launched by pressing "…".

Let's consider a parameterization row example: `Text Area 0@$(title)@PIP Frame 0@$(gpi1_val)`. This row sends parameters into two objects of a graphic composition: `Text Area 0` and `PIP Frame 0`. The first object will receive the title of a clip played in the main playlist. The

second one will receive a clip with name set in the GPI V 1 field of the main playlist.
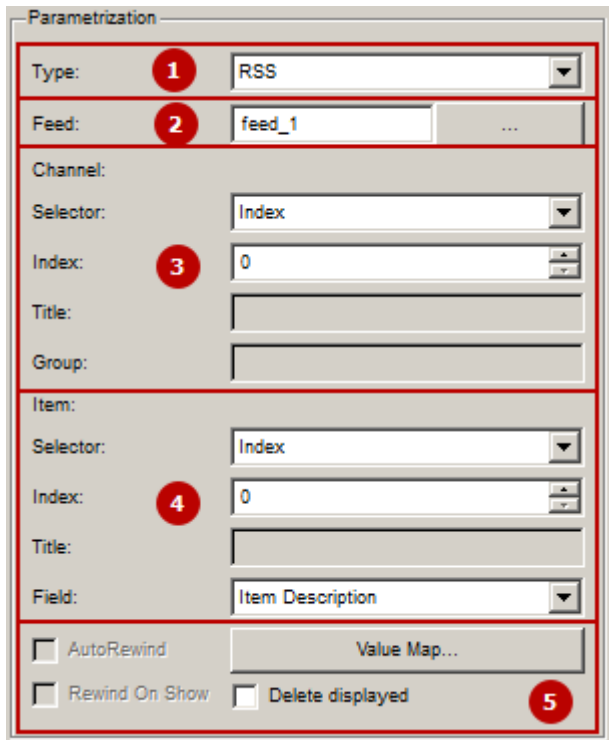
RSS Parametrization

In this mode, data is loaded from the RSS channel by Parser, built-in to parameterized elements. The RSS channel page must be accessible in the network from all computers that interact with the graphic composition (graphic station, broadcasting server, etc). Sending data in the RSS format is available to, for example, SMS-providers and meteostations.
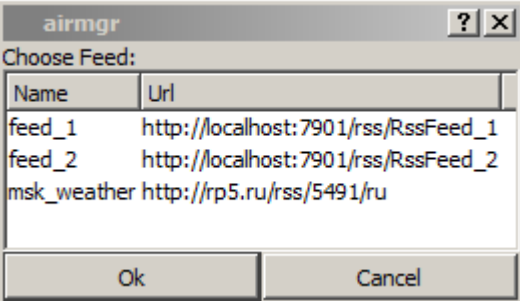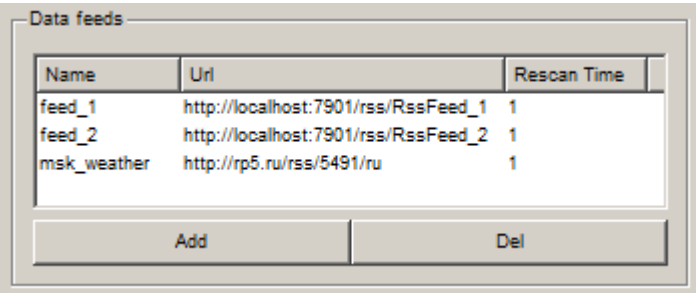
Parametrized Objects

RSS parameterization is available to the following objects:

Object	Description	Usage
* **PIP frame**	Parameterization used to specify a displayed clip: a clip name from the media database or a network resource URL. Supports loading images (PNG, GIF, JPEG) and video (MP4) from network. Loading data to the object is implemented once at the moment of playback start, displaying data from the first available <item> element. To move to the next element, you must reload the object with object_hide and object_show commands via Graphic API or restart the whole composition.	Used for output of weather icons, traffic lights, etc.
Text object	Parameterization used to load a displayed text. Loading data to the object is implemented once at the moment of playback start, displaying data from the first available <item> element. To move to the next element, you must reload the object with object_hide and object_show commands via Graphic API or restart the whole composition.	Used to output text information that does not require being updated during playback: "Now Playing" Captions, "Coming Up Next" Captions, age markers, etc.
Text feed	Parameterization used to load a displayed text. The Text Feed object consistently displays all available <item> elements. RSS feeds are regularly checked for updates. New elements are added to the playback queue.	Used for creating tickers, displaying weather and other information that may change during playback of a graphic composition.

Activation and Configuring

Selecting the RSS type of parameterization activates the RSS-parser built-in to the graphic object. Its settings may be divided into 5 parts filled in consistently:

Parameter	Description
1 – selecting type	
Type	Parameterization type: RSS, External and Disabled.
2 – selecting source	
Feed	Select a configured RSS-feed from the list: Connecting feeds to the composition is implemented in Graphics Editor, at the General tab: • **Name** – feed internal name • **Url** – link to the RSS-feed on the network • **RescalTime** – resource polling interval
3 – settings for selecting channel(s) inside the RSS-feed	

Parameter	Description
1 – selecting type	
Selector	Defines a condition for selecting a finite subset of channels that meet the set criteria from all channels in the RSS-feed. Note: RSS standard defines one channel per an RSS-feed. However, the Skylark server can process streams different from the standard, containing several channels. • **Index** – selecting one channel by its sequence number set in the Index field. • **Title** – selecting one channel by its title set in the Title field. • **Feed** – consistent searching through all channels in the feed that meet the Group condition.
Index	Defines the number of the selected channel (enumeration from 0). The option is active if Selector=Index.
Title	Defines the title of the selected channel, active when Selector=Title. The field content must be completely similar to the `<title></title>` content of the selected channel.
Group	The option is active if Selector=Feed.
4 – Selecting data from the elements of a chosen channel(s)	
Selector	Defines a condition for selecting a finite subset of channels that meet the set criteria from channels selected at the previous stage. • **Index** – selecting one `<item></item>` element by its sequence number set in the Index field. • **Title** – selecting one `<item></item>` element by its title set in the Title field. • **Feed** – consistent searching through all `<item></item>` elements in the feed. • **Channel Description** – output of content of the channel descriptive tag `<channel><description></description></channel>`.
Index	Defines the number of the selected element (enumeration from 0), active when Selector=Index.
Title	Defines the title of the selected element, active when Selector=Title. The field content must be completely similar to the `<title></title>` content of the selected element.
Field	Defines the `<item></item>` tag with RSS-element data that must be displayed. • **Item Description** – displays content of the `<description></description>` tag • **Item Title** – displays content of the `<title></title>` tag • **Item Enclosure** – displays content of the `<enclosure></enclosure>` tag
5 – additional settings	
AutoRewind	While displaying RSS data, the system remembers GUID of displayed `<item>` RSS-elements. The feature launches displaying RSS-elements all over again and resets the history when the feed is out of elements.
Rewind On Show	While displaying RSS data, the system remembers GUID of displayed `<item>` RSS-elements. The feature launches displaying RSS-elements all over again and resets the history after restarting the graphic composition in the playlist.
Delete displayed	🔧 **Fix Me!**

Parameter	Description
1 – selecting type	
Value Map	Replaces words or phrases in the text from the element field.

Example of RSS Stream from RSS Feeds

The RSS Feeds server component forms different RSS streams, depending on the input data format. On of the possible variants:

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
    <!-- Channel #0 -->
    <channel>
        <title>news100_channel</title>
        <!-- Channel description; element content is displayed, when
Selector=Channel Description -->
        <description>Lead Story</description>
        <!-- item #0 element -->
        <item>
            <!-- Element content is displayed, when Field=Item Title -->
            <title>100_line_2</title>
            <!-- Element content is displayed, when Field=Item Description -
->
            <description>Eight people hospitalized after a bus accident near
Krasnodar</description>
            <guid>Z:\MAM\RSS/100.txt_line_2</guid>
        </item>
        <!-- item #1 element -->
        <item>
            <title>100_line_3</title>
            <description>Largest deal in history of aviation completed at
the air show in Dubai</description>
            <guid>Z:\MAM\RSS/100.txt_line_3</guid>
        </item>
        <!-- item #2 element -->
        <item>
            <title>100_line_4</title>
            <description>State Duma Budget Committee recommended rejecting
the law on Rosfinagency establishment</description>
            <guid>Z:\MAM\RSS/100.txt_line_4</guid>
        </item>
    </channel>
    <!-- Channel #1 -->
    <channel>
    ...
    </channel>
<!-- Канал №N -->
...
</rss>
```

Link to this RSS-feed.

> The RSS standard defines one channel per an RSS-feed.
> However, the Skylark server can process streams different
> from the standard, containing several channels.

Example of RSS News Feed

A demo RSS from the portal Rambler is used as an example.

```
<?xml version="1.0" encoding="UTF-8"?>
    <rss xmlns:rambler="http://news.rambler.ru" version="2.0">
    <channel>
        <title>Title of your website</title>
        <link>http://www.site.ru/</link>
        <description>Short description of your website</description>
        <!-- Basic element -->
        <item>
            <guid isPermaLink="false">2334456</guid>
            <title>News title</title>
            <link>http://www.site.ru/news/2334456/</link>
            <pubDate>Mon, 09 Feb 2009 03:03:21 +0400</pubDate>
            <description>Short description</description>
            <category>Category</category>
            <rambler:fulltext><![CDATA[News full text.]]></rambler:fulltext>
            <author>Author</author>
            <!-- The element content is displayed, when Field=Item Enclosure
-->
            <enclosure url="http://images.site.ru/12341234.1234234.jpg"
type="image/jpeg" length="123" />
        </item>
        <item>
            ...
        </item>
        ...
    </channel>
</rss>
```

Link to this RSS-feed.

Disabling Object Parametrization

Open the parameterization setting menu and set the Type switcher to Disabled.

From:
https://wiki.skylark.tv/ - **wiki.skylark.tv**

Permanent link:
https://wiki.skylark.tv/manual/graphics_editor_parametrization

Last update: **2020/06/17 10:58**