Reference Manual

GR*i*NS Editor for RealOne and **GR*i*NS Pro Editor for SMIL 2.0**
Windows-98/2000/XP
February, 2002

GR*i*NS Editor Reference Manual for Windows-98/2000/XP. February, 2002.

Important Notices

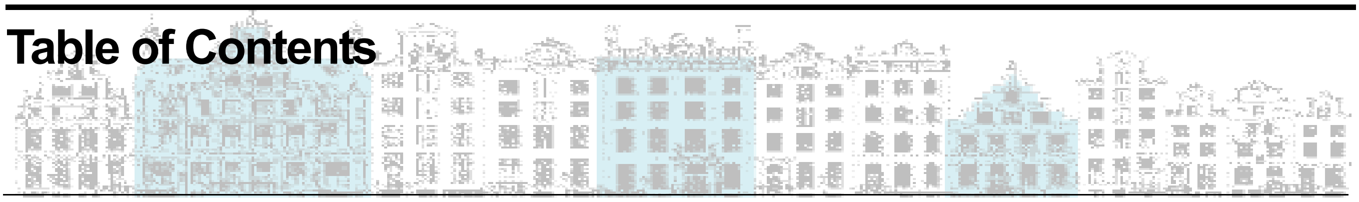
This document is the GR*i*NS Editor *Reference Manual*. All of the information presented in this publication has been verified, but incremental product updates may impact part of this guide.

This version of the GR*i*NS Editor *Reference Manual* has been produced for use as an off-line reference. Images and page layout have been optimized for printing on a 600-dpi (or greater) laser printer. For best reproducibility, the use of a color printer is recommended, although every effort has been made to make illustrations readable on other printers as well. If you wish to use it as an on-line reference via a PDF reader, we recommend that you increase the level of display magnification when viewing images.

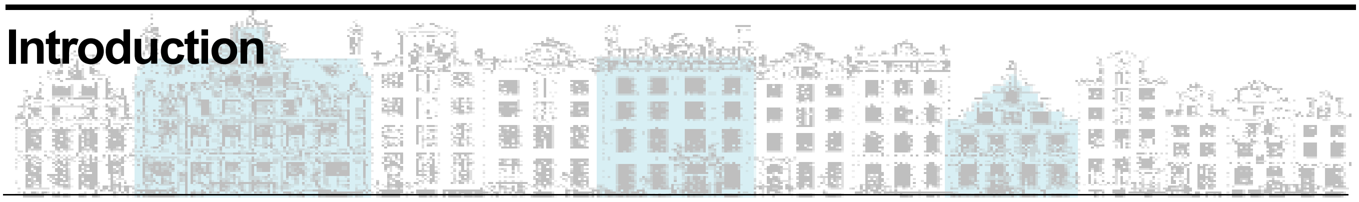
The images used in this publication were taken from the GR*i*NS Editor for RealOne build 2.0-win32-136 for Windows 2000 and/or GR*i*NS/Pro build 2.2-win32-135. While the look of other versions of GR*i*NS are slightly different because of adherence to common conventions on those other environments, the functionality described is similar for all versions of GR*i*NS. In order to reduce document size, only images from the Windows version have been included in this document.

We welcome your questions on GR*i*NS Editor for RealOne and comments on this documentation. Please submit all questions and comments to our support desk at *grins-support@oratrix.com*. We maintain a list server dedicated to sharing experiences among GR*i*NS Editor for RealOne users. See the on-line release notes that come with the software distribution for details of this listserver. Finally, if you wish to submit your own SMIL files as examples for other users, please send a request for submission to: *grins-examples@oratrix.com*.

Table of Contents



Introduction



About this manual

This manual is a reference document that explains the functionality of the core features of GR*i*NS. It is not intended to be tutorial in nature. Readers should also consult other GR*i*NS documentation:

- the *Quick Start Guide*, which contains installation and initial use instructions;
- the *Templates Guide*, which describes the basic GR*i*NS template set;
- the *HowTo Guides*, which gives a step-by-step overview of how GR*i*NS can be used to solve particular design tasks.

Readers should also consult relevant documentation for each of the players to be used: the RealOne IQ Guide (for the RealOne player), Microsoft's Knowledge Base (keyword: time2) (for the IE-6.0 HTML+TIME SMIL implementation), the 3GPP consortium (for 3GPP/Wireless+Mobile SMIL) and, as a final authority, the official W3C SMIL 2.0 Specification (available at: **http://www.w3.org/AudioVideo/**). Note that all of the SMIL 2.0 functionality provided by the GR*i*NS editor is also available in the GR*i*NS family of players.

Published Formats

GR*i*NS documentation is available in the following formats:

PDF The PDF version, to be used with the Adobe Acrobat 4.0 and later viewer, is intended as an off-line reference document, suited for printing on a 600-dpi or greater printer.

HTML An HTML version, installed in the GR*i*NS home directory, suited for on-screen viewing.

Program Interface Contents of this Reference manual will be linkable through the GR*i*NS help system (via the Help pull-down menu). This requires a connection to the Internet as it accesses the latest in documentation.

Note that not all versions may be available in early product releases.

Text Formats

A series of type faces are employed to distinguish information groups.

Code Text used to show SMIL coding is in a Courier format. `This is an example.`

Hyperlink Hyperlinks are used to navigate to other documents. They are bold, blue and underlined. *This is an example*.

IMPORTANT

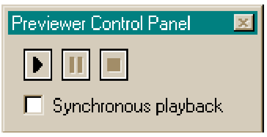
The GR*i*NS editor comes in a variety of configurations. This Reference Manual describes all of the features available in all of the GR*i*NS implementations. Where appropriate, functions that are only available in limited configurations will be labelled. Unless otherwise stated, functionality is available in all GR*i*NS versions.

General User Interface

Toolbars

The GR*i*NS user interface provides a set of control panels and toolbars that contain short-cut icons for common operations. Toolbars can be anchored to the GR*i*NS main window or floating. Toolbar visibility is controlled through the View->toolbars pull-down menu. See also “Toolbars” on page 13.

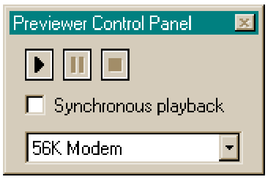
Previewer Control Panel



Previewer Control Panel

☐ Synchronous playback

The previewer control panel controls the GR*i*NS internal preview of a document. It always has buttons for the Play, Pause, and Stop commands. You may also select control over synchronous playback: if selected, the value of the SMIL syncMaster attribute is used to control synchronization during playback in the Previewer.

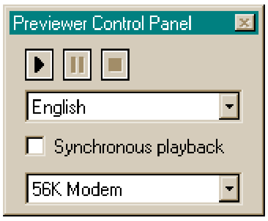


Previewer Control Panel

☐ Synchronous playback

56K Modem

If Bandwidth Use Monitoring has been selected from the View menu, the Previewer will evaluate all Bitrate switch settings with the displayed value.



Previewer Control Panel

English

☐ Synchronous playback

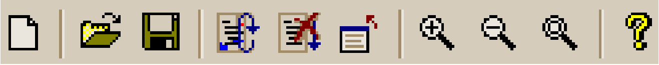
56K Modem

If one or more nodes have the System Test *Language* attribute set, the Previewer will evaluate all switch settings with the displayed value.

Note: GR*i*NS/RealOne allows *systemLanguage* and *systemBitrate* to be set. GR*i*NS/Pro allows all SMIL test attributes to be set/evaluated, including *systemCaptions*.

File and General

This toolbar contains single-click buttons to the File menu items:



New Create a new presentation.
See “New...” on page 7.

Open Open an existing presentation.
See “Open...” on page 7.

Save Save the presentation.
See “Save” on page 7.

Restore Revert to the last saved version.
See “Revert to saved” on page 8.

Close Close the current document.

Close window Close the current view or dialog.

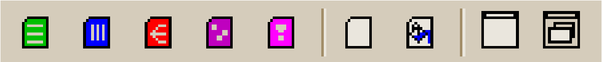
Zoom in/out Zoom the timeline or the layout view. See also “Show Timeline/ Bandwidth” on page 13.

Help Short-cut help icon. (For future use.)

Containers

This toolbar contains items that can be dragged to the structured timeline to create a container (group) node or media object. The same result also can be achieved through menubar commands or using the right-click contextual menus. All of the icons below are also available in the Insert pull-down menu.

The container icons are active only when in the structured timeline. For more details about the behaviors of these objects "Containers (Groups)" on page 17.



Par Create a Parallel container; all children share a common time base, starting at the beginning of the parallel.

Seq Create a Sequential container; each component is scheduled after its predecessor.

Switch Create a Switch container; at most one of the children will be activated, based on Language or Bitrate settings on the items.

Excl Create an Exclusive container; at most one child will be active, based on event behavior or scheduled start times.

Priority Class Create a Priority container; define a priority grouping for contained nodes. *(May be used inside an Excl only.)*

Media Item Create an empty Media object; the media node's properties will determine when and how it is rendered.

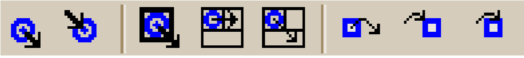
Brush Create a Brush object; fills an area with the color based on the node's properties.

The following icons are only active when in the Layout view:

Top Layout Create a new Top-Layout. "Top-layout" on page 21. *(Note:* GR*i*NS/Pro *Only.)*

Region Create a new region that is a child of the parent region/layout. "Region" on page 21

Timing and Linking



This toolbar allows synchronization and timing events to be added to containers and media objects in any view where presentation objects (media, nodes) can be selected. They are single-click buttons that apply to the currently selected object. Only those icons that are relevant to the current editing context are active. For instance, a begin event or hyper link destination cannot be created unless the source has been created and selected first.

Internal Link Source Use the selection as the source of a hyperlink to an internal node. (GR*i*NS/Pro Only; not supported by HTML+TIME.)

Internal Link Destination Use as the destination of a hyperlink to an internal node. (GR*i*NS/Pro Only; not supported by HTML+TIME.)

External Link Use the selection as the source of a hyperlink to an external node. See “Create full link source and edit” on page 11.

Context Link Use the selection as the source of a trigger to place content in the RealOne *Context* window. “Create link to context window...” on page 11. (Only supported by RealOne player.)

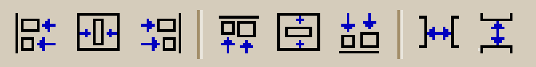
Browser Link Use the selection as the source of a trigger to place content in the RealOne *Browser* window. “Create link to browser window...” on page 11. (Only supported by RealOne player.)

Event source Use the selection as the source of an event. “Use as event source” on page 11.

Create begin event Create a begin event on the selection using the previously selected event source. “Create begin event” on page 11.

Create end event Create an end event on the selection. “Create end event” on page 12.

Align / Distribute (GR*i*NS/Pro Only)



These single-click buttons are active when two or more objects are selected in the layout view (“Layout Overview” on page 21) for purpose of physical arrangement.

The *align/distribute* toolbar is active only for the Layout view and is only available in the GR*i*NS/Pro product.

Left edges Align left edges of 2 or more items.

V-Centers Align vertical centers of 2 or more items.

Right edges Align right edges of 2 or more items.

Top edges Align top edges of 2 or more items.

H-Centers Align horizontal centers of 2 or more items.

Bottom edges Align bottom edges of 2 or more items.

Distribute horizontal Divide available horizontal space between 3 or more items.

Distribute vertical Divide available horizontal space between 3 or more items.

Pull-Down Menus

General Information

All commands in the GR*i*NS Editors are available via the menu bar. (The only exceptions to this rule are commands that have meaning only in the context of a single dialog.)

Many common commands are also available in a contextual popup menu on the object, invoked with the right mouse button, but these contextual menus are only a subset of the full command set available in the menu bar.

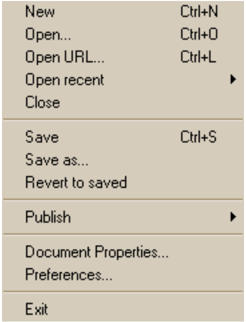
Many commands also have keyboard shortcuts which can be seen in the pull-down menu, or toolbar buttons (See “Toolbars” on page 3), but again these are a subset of the complete command set.

Commands are active only if they make sense given the current selection, state of the document, etc.

Pull-Down Menu Sets

File Menu

The structure for the File menu is shown at right.



New... Create a new document from a template. (The template selection dialog is shown as a result of this command.) A filename will be requested upon creating the new document.

Open... Open an existing presentation. The presentation can be a GR*i*NS project file (*.grins* or *.xgrins*) or a standard SMIL file (*.smi* or *.smil*).

Open URL... Opens an existing presentation by giving a URL, allowing you to read in presentations from a web server. However, since you cannot save the presentation back to the web, so you will be prompted to use Save as....

Open recent A submenu of the most recent presentations edited, for quick access.

Close Closes the current presentation and all its views.

Save Save the current presentation.

Save as... Save the current presentation to a new *.grins* file or *.xgrins* binary-coded presentation file. Use Publish to produce a *.smil* file.

Revert to saved Discards all the changes made to the document since the last save.

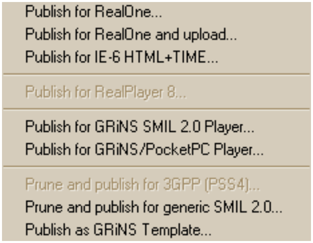
Publish A submenu providing exports of a GR*i*NS presentation. (See

Document Properties This shows the document properties dialog (See “Info Tab” on page 43). Top-level document properties can be set such as Author and Copyright information, as well as FTP addresses for uploading to the server that will host the data.

Preferences... Document specific preferences that allow the author to set options such as synchronous playback or template creation. (GR*i*NS/Pro Only.)

Exit Exit the GR*i*NS Editor session.

Publish Sub-Menu


Publish for RealOne...
Publish for RealOne and upload...
Publish for IE-6 HTML+TIME...
Publish for RealPlayer 8...
Publish for GRiNS SMIL 2.0 Player...
Publish for GRiNS/PocketPC Player...
Prune and publish for 3GPP (PSS4)...
Prune and publish for generic SMIL 2.0...
Publish as GRiNS Template...

The structure of the File->Publish menu is shown at right. Note that as part of presentation publishing, media objects are copied into a single distribution directory and a file name prompt is given.

Publish for RealOne ... Takes the current project and creates a *.smil* file that is compatible with the RealOne player. (Media objects may be converted as part of the publish; see “Media Conversion” on page 48.)

Publish for RealOne and upload ... Same as *Publish for RealOne*, except that the resulting presentation is uploaded to a server based on the properties defined for the project. See “GRiNS Preferences” on page 40.

The following commands are active when using the GR*i*NS/Pro Editor only. (Note that not all commands are supported in all versions.)

Publish for IE-6 HTML+TIME Converts the syntax of the project to the XHTML+SMIL profile, as adapted by Microsoft for HTML+TIME. (Not all SMIL constructs are supported by IE-6; GR*i*NS/Pro Only.)

Publish for RealPlayer 8 Converts the presentation to a SMIL 1.0 format, as supported by RealPlayer 8. (Special editions of GR*i*NS/Pro Only.)

Publish for GRiNS SMIL 2 Player ... Creates a presentation in full SMIL 2.0 (Language Profile) compliance format. Will work with any fully compliant SMIL 2.0 player, such as the GRiNS/SMIL-2.0 Player. (GR*i*NS/Pro Only.)

Publish for GRiNS / PocketPC Creates a presentation that is compatible with the GRiNS scalable PocketPC Player. (GR*i*NS/Pro Only; special versions support device enahancements.)

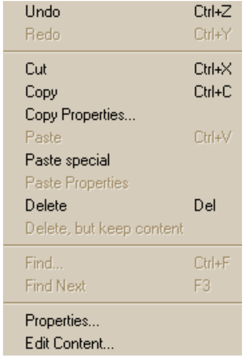
Prune and Publish for 3GPP/PSS4 Creates a presentation that is compatible with the 3GPP Mobile SMIL profile (version PSS4). Will remove any options and conditional content not compatible with that profile. (GR*i*NS/Pro Only; special versions support device enhancements.)

Prune and Publish for SMIL 2.0 Similar to Publish for GRiNS SMIL 2.0 Player, except that it will remove any options and conditional content not compatible with that profile. (GR*i*NS/Pro Only.)

Publish as GRiNS Template Allows the current project to be saved in a template format, with all template settings enabled. See also “Per-Object Template Setting” on page 54. (GR*i*NS/Pro Only.)

Edit Menu

The structure of the Edit menu is shown at right:



Undo Undo the last edit operation. Edit operations are operations that modify the presentation (or the clipboard); non-editing operations, such as selecting a different node are not undoable. The undo stack is accessible until saving the document.

Redo Redo the operation most recently undone.

Cut Remove the currently selected object and place it on the clipboard. References to and from the item on the clipboard (begin events, hyperlinks, etc.) will continue to point to that object.

Copy Create a copy of the selection and put this copy on the clipboard.

Copy properties Shows a dialog that lists all properties on the currently selected item. Highlight the desired properties and they are copied to the clipboard. This is very useful for quickly duplicating special properties to many places (such as duration or region placement).

Paste Put the contents of the clipboard back in the document, using *paste after* semantics.

Paste special Similar to paste but with an object placement submenu that allows *paste before*, *paste within* and/or *paste after* semantics.

Paste Properties Used to paste the *Copied Properties* onto the currently selected object.

Delete Remove the currently selected object from the document.

Delete, but keep content Deletes the container currently selected, but leaves the media objects that were contained within the container in the parent container.

Properties... Show the properties dialog for the current selection. In the properties dialog you can set such things as duration (for media items) or background color (for regions). See “Property Dialogs” on page 37 for more information.

Edit content Open an external media editor for the media item in the current selection. This only works if the item resides on the local disk, and if a media editor is known for the specific media type.

The following commands are active during editing in the source view. (GR*i*NS/Pro only.)

Find... Search for text.

Find next Find the next occurrence of the search string.

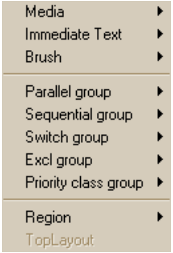
Replace... Search for text and replace it.

Insert Menu

Insert commands are used for inserting new containers and objects and regions into a presentation.

The insert media commands have three varieties that dictate where the new object will be inserted. These positions are *before*, *after* and *within*. The latter is only available if the selection is a container such as a sequential or parallel node.

The structure of the Insert menu is shown at right:



Media Insert an empty media item. Use drag and drop or the property dialog to select the media file.

Immediate text A media item that will hold an amount of text. This is a convenience item for small text items such as captions.

Brush A media item that puts a solid color in a region.

The *insert container* commands have an additional sub-command *parent* which creates the new container node as the parent of the current selection. When selected, a dialog is shown highlighting properties of the current selection that will propagate (move) to the new parent node. The remaining properties are left on the child node. This can be used to ensure that certain events and attributes continue to function as desired.

Parallel A container item that plays all its children (the items contained within itself) relative to the same time base and (by default) finishes playing when all its children are finished.

Sequential An item that plays its children in sequence, with each successor scheduled relative to its predecessor. The seq ends when the last child is done or when the container itself has a set duration.

Switch A group that selects one of its children, based on system test attributes such as the bitrate of the internet connection.

Exclusive A group that plays at maximum one of its children at any given time. *Exclusives* can be used to control presentation flow by event-based user interaction.

Priority Class A sub-group of the *Exclusive* node. Each priority class defines a priority container with contents that can interrupt a low-priority exclusive member. The interruption can be temporary or permanent, based on priority class attributes.

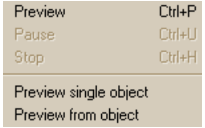
Region Insert a new region within the current selection.

Toplayout Create a new main window in which regions can be created. (GR*i*NS/Pro only.)

Preview Menu

These commands pertain to the integrated GR*i*NS/SMIL-2 preview player. Some are also available from the right-click contextual menus and from the Preview Control Panel.

The structure of the Preview menu is shown at right:



Preview Start playing the presentation from the beginning.

Pause Pause the presentation.

Stop Stop playing the presentation.

The Preview menu also allows you to selective preview part of a document. This can be a single node, such as a media node, or a group of nodes.

Preview single object Play the current selection only. This is only meaningful for nodes (either single media or a sub-group) that resolves to an explicit duration.

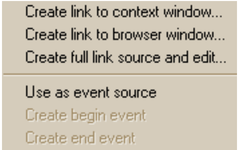
Preview from object Play the presentation starting at the current selection.

Custom tests Select and preview the document using the custom tests that are specified in the document. (GR*i*NS/Pro only; only available when using the GR*i*NS family of players or other players with full SMIL 2.0 Language support.)

Linking Menu

These commands are specific to the interaction of objects and timing of the document. They are available only in the structured timeline. (See the notes at the end of this section for important information on linking functionality.)

The structure of the Linking menu is shown at right:


Create link to context window...
Create link to browser window...
Create full link source and edit...
Use as event source
Create begin event
Create end event

Create link to context window... Creates a link to an HTML page that is to be displayed in the RealOne context window. The content is displayed without user interaction.

Create link to browser window... Creates a link to an HTML page that is to be displayed in the RealOne browser window. The content is displayed without user interaction.

Create full link source and edit Use the currently selected media item as the source of an external hyperlink.

Use as event source Use the currently selected item as the source of a begin or end event. By default this creates an event triggered by the viewer clicking the media item. This can later be changed in the property dialog when the event is completed.

Create begin event Create a begin event on the currently selected item. The item will only start to play when the event is triggered. This function is only active when an event source has been created previously. Details of the event, such as the activation method, such as delay can be modified through the node properties (See “Begin/End Tabs” on page 43).

Create end event Similar to *Create begin event* but an end event is created on the currently selected item. The item will stop playing when the event is triggered.

The following pull-down menu entries are available on in the GR*i*NS/Pro Editor:

Create simple link source Creates a the source for a hyperlink that can be used to reference an internal or external document or object.

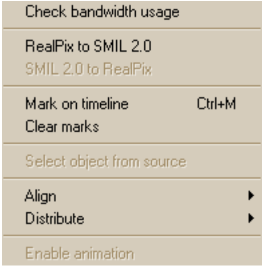
Finish simple link source Creates the destination to the source of a simple link.

***Important Notes*:**

1. When defining Context and Browsers links in documents that are rendered with players other than RealOne, the associated Web content is displayed in the operating system's default HTML player.
2. When publishing to HTML+TIME, be aware that the IE-5.5/6 implementations do not support hyperlinks to any SMIL 2.0 construct. A warning is generated on publish.

Tools Menu

The structure of the Tools menu is shown at right:



Check bandwidth usage Checks that the bandwidth usage by the media elements in the document is consistent with the system bitrate as set in the player control panel (See "Previewer Control Panel" on page 3). A short report is shown giving presentation startup delay (preroll time) and the number of times the presentation will pause and rebuffer (stall). Individual media items will get a bandwidth icon that shows whether the item caused any bandwidth problems (See "Show Bandwidth" on page 20).

RealPix to SMIL 2.0 Convert a RealPix media item to the corresponding sequence of images and transitions using SMIL 2 and RealOne syntax. This command is used when converting legacy RealPlayer documents to SMIL 2. (Currently supported as Beta functionality.)

Enable animation Active only in the Layout view if the current selection is a media item. Allows animation of the items position, size, background color, etc. See "Animation" on page 23 regarding using the animateNode facility.

Mark on Timeline Allows a marker to be placed on the timeline during playback. A typical use-case is to place markers on the timeline during playback of an audio object; images or text captions can be synchronized to the audio by aligning them with the markers. (Note: works best when previewing a single node. May not function correctly if timeline discontinuities exist. See also: "Time, Structure and Bandwidth" on page 20. Note that Marks are temporary objects that will be reset on close and open.)

Clear marks Remove all markers from the timeline.

The following pull-down menu entries are available only in the GR*i*NS/Pro Editor:

Align Aligns two selected objects together according to the option selected.

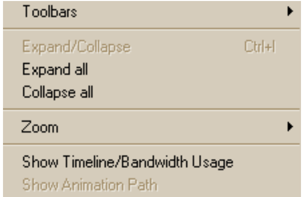
Distribute Distribute two objects with equal horizontal or vertical space. The outermost two elements stay where they are, the other elements are positioned so that all elements are equally spaced. This command works best if all items are the same size and non-overlapping.

SMIL 2.0 to RealPix The reverse of *RealPix to SMIL 2.0* where a sequence of images can be converted to a single base64 encoded RealPix object. (GR*i*NS/Pro only; currently supported as Beta functionality.)

Select node from source Active only in the source view. This focus in the StructuredTimeline is set to the object containing the text cursor in the source view. (GR*i*NS/Pro only.)

View Menu

The structure of the View menu is shown at right:



Toolbars Shows or hides the various toolbars and the player control panel. See "Toolbars" on page 3 for an explanation of the individual toolbars.

Expand/ Collapse Expand or collapse a container item, showing and hiding its contents. Useful for complex presentation to un-clutter the structure view.

Expand all Expand the selected container item and all collapsed child container items.

Collapse all Collapse the selected container item and all container items contained within it.

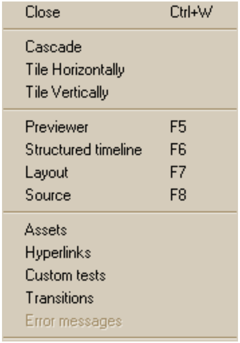
Zoom In the timeline-enabled structure view, this enlarges or diminishes the timeline. In the layout view it magnifies the preview canvas. (A rollover menu provides a zoom factor selector.)

Show Timeline/ Bandwidth Toggle the display of time in the structure view. Showing the timing of your media objects in a display that is governed by length of time rather than purely by structure. It is particularly useful for adjusting media durations and delays. Not showing time can display more items on the screen at the same time, and is the preferred mode of viewing when editing the high-level structure of a presentation. *Show time* also turns on the begin/duration drag handles for objects in the structure view (See "Show Time" on page 20).

Show Animation Path Available only in the Layout view, shows a dotted line that represents the path in which a media object is animated. The path follows the top-left point of the animated media object.

Window Menu

The structure of the Window menu is shown at right:



Close Closes the topmost view or dialog window.

Cascade Place windows independently on the screen.

Tile horizontally Stack all open views horizontally. Note that this also tiles the previewer window, if open.

Tile vertically Stack all windows vertically.

The use of tiled window configurations is not recommended because of its interaction with the Previewer. It is supported for legacy reasons only.

The following options select a GR*i*NS editing view:

Previewer view Shows the previewer window.

Structured Timeline Shows the structured timeline, the main editing view of GR*i*NS (See "Structured Timeline" on page 17).

Layout view Show the layout view, where region layout, placement of media and animation editing occurs (See "Layout" on page 21).

Source view Show the source view, where the raw smil source is viewable in XML format (See "Source View" on page 27). *Source editing is available only in the* GR*i*NS/Pro *product.*

The following views are for special-purpose editing windows.

Assets Show the assets view, where an index of used/unused media items and code snippets are kept for re-use in the document (See "Source View" on page 27).

Hyperlinks Shows the hyperlink editor, which allows navigation and creation of advanced hyperlinks (See "Hyperlink View" on page 31).

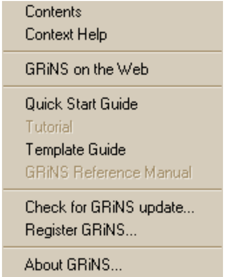
CustomTests Allows the definition and manipulation of SMIL's Custom Test Attributes. (GR*i*NS/Pro only; only supported by the GR*i*NS family of players.)

Transitions Shows the transition view, which allows creation and modification of media transitions (See "Transitions View" on page 25).

Error messages Opens a subwindow that is available only when the document currently opened contains errors in the SMIL code. (See "Source File Error Messages" on page 35.) (GR*i*NS/Pro only.)

Help

The structure of the Help menu is shown at right:



Contents Opens a web-browser which connects to the GR*i*NS online documentation set. This directs to the main index for the documentation regarding the particular product in use.

Context Help Opens a web-browser which connects to the GR*i*NS online documentation set. This directs to the informative help regarding the view which is currently in the foreground such as the structure view, layout view, etc. *(Available in* GR*i*NS*/Pro only.)*

GR*i*NS on the web Opens a web-browser which connects to the homepage of the Oratrix web site.

Quick Start Guide Accesses the Quick Start Guide that is installed to the local machine. Viewing requires Adobe Acrobat.

Tutorial Accesses the Tutorial document that is installed to the local machine. Viewing requires Adobe Acrobat.

Template Guide Accesses the Template Design Guide that is installed to the local machine. Viewing requires Adobe Acrobat.

GR*i*NS Reference manual Accesses the GR*i*NS Reference manual that is installed to the local machine. Viewing requires Adobe Acrobat. (You are currently reading this manual!)

Check for GR*i*NS update... Opens a web-browser which connects to the Oratrix web site to check for a newer version or upgrade to the particular product in use.

Register GR*i*NS... Opens a web-browser which connects to the Oratrix web site to register the product. This is only appropriate for license-holders who have purchased their copy from an official reseller of GR*i*NS products, such as Real Networks. Those who have purchased from Oratrix will not need to register.

About GR*i*NS Displays information on the version and build of the GR*i*NS product

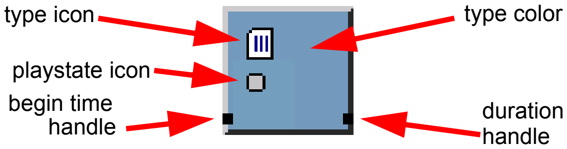
Structured Timeline

Overview of the Structured Timeline

The structured timeline is the main editing interface for GR*i*NS. The following text outlines the fundamentals of using this view.

Each object in the Structured Timeline is identified by a color and a type icon. The state of the object is also identified by the color of its playstate icon.

The following image identifies the main components of a node in the structure view:

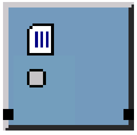


In addition, media items also have special visual properties, to be described below.

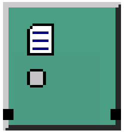
Containers and Icons

Containers (Groups)

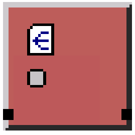
Containers are structure buckets that hold a series of objects. The type of container dictates how child objects are scheduled and activated. The structured timeline uses a variety of visual concepts to denote the playable structure of a document.



Sequential All children of the sequential container are played one at a time in the order that they appear in the container. Scheduling within a sequential container is *relative*.

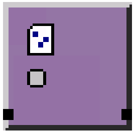


Parallel All children of the parallel container are played according to the same time base. Scheduling within a parallel container is *absolute with respect to the start of the parallel*.



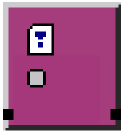
Switch A switch container defines a collection of one or more children, of which at most one will be activated. (It is also possible that none of the children are activated.) Each child of the switch must have either a system test attribute that triggers the playing of that one child. In GR*i*NS/RealOne, these attributes may only be a language and/or bitrate setting. In GR*i*NS/Pro, all possible system test and custom test attribute settings are allowed. When the switch is played, the first child that fits the profile of the setting is activated.

Exclusive



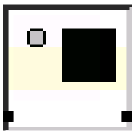
The exclusive container is used for manual interaction in a presentation. Children of the exclusive container are played only if they are referenced by a parent object of the exclusive container. This allows user control of the presentation such as menu selected navigation.

Priority



Priority containers determine the relative priority among a set of priority class containers in an exclusive. If a priority class is used by one child of an exclusive, all children of that exclusive container must be priority containers. The priority class provides a layer of logic that deals with interrupt and pause actions among the children of the exclusive. One practical application is the use of banner ads in a presentation that should change when users are interacting with higher-level parts of the document. For the most part, these priority class containers are timeline transparent.

Brush



This is a pseudo media item containing a color, a layout region reference and a duration. The color of the box in the icon shows the color of the brush element.

Information/Interaction Icons

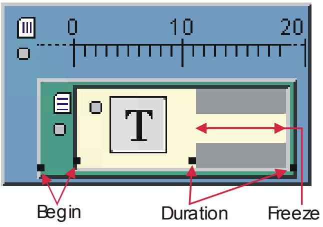
Each container is displayed with a set of icons that give feedback on activation and state information. The purpose is to visually aid content-creation rather than bog the user down with endless text strings.

Playstate When playing a document, the playstate icons feedback the current status and position of the playing. When user interaction in a presentation occurs, certain containers are played, others are ignored. The three playstate indicators are as follows:

- **Playing**: When a parent container or media object is currently being played, the playstate indicator illuminates a bright green.
- **Skipped**: If a parent container or media object did not play for any reason, it turns grey to show it was skipped. This is very useful for showing errors an navigational choice in a presentation.

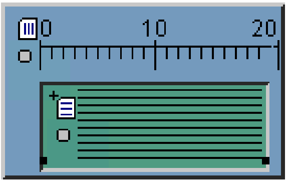
Drag handles Drag handles allow the (relative) begin time and forced duration of an object to be specified graphically. The right handle governs duration and the left governs the begin offset. In *constrained mode*, you may only alter the duration of an object between the predecessor's end and the successor's begin. If you depress the Ctrl key while moving the drag handle, you may push back successor objects during drag operations.

Freeze field If a node has a *fill* attribute with a value of *freeze*, a grey/yellow background is drawn at after the end of the active duration of the object. The



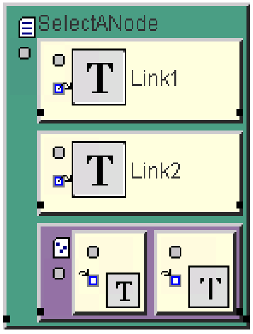
yellow bar against the grey background represents the duration that the object remains on the screen after its scheduling duration has ended.

Expand/ Collapse



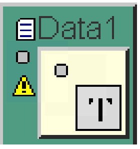
Each type icon in the top-left corner also serves as an expand/collapse toggle. Clicking on the icon will show or hide the contents of that container, allowing more detail, or more work space. This is useful for complex presentations that often get visually cluttered, especially when work is being done on a different part of the document. Closing containers can also improve the general performance of the Editor with manipulating large documents.

Events



Event icons serve two purposes. They denote the presence of an event, and the connection of that event. When creating an event, and finishing that connection, the two objects involved have the icons that indicate the presence of the event, and then clicking on one of those icons will show the connection that was made and the direction.

Warnings



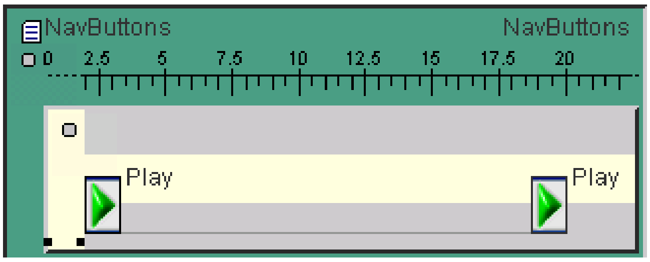
Warning icons notify the user that an error has occurred in the presentation. They appear when the document is played and the error has occurred. They most often denote that an object will not be played because of attribute problems. Clicking the warning icon will show the error message.

Time, Structure and Bandwidth

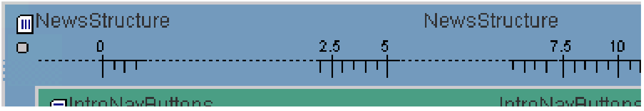
Show Time

Show time in a presentation enables the timeline and is enabled by default. The timeline is used in all time-based authoring, and adopted into the structured timeline with the same purpose. However in GR*i*NS, the timeline must have additional behavior to fit non-linear timelines as complex presentations most often have. Some exceptions apply for when the timeline is broken into segments because of gaps required to draw presentation structure.

The following image depicts a standard timeline:

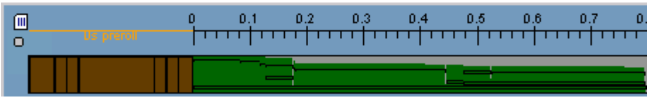


When special-class containers are used in a presentation (particularly exclusive and priority class), or when the presentation contains large numbers of embedded structure containers, breaks in the timeline may occur. In this case, the timeline breaks into pieces called *timeline discontinuities* — these are intervals that take up space but do not take up any time in the presentation.



(Note: interactive event-based timing is ignored for the timeline; objects that start on events are shown as started at the time the event is evaluated.)

Show Bandwidth



The bandwidth strip computes expected bandwidth usage and displays it underneath the main timeline. The computation is done based on the bitrate set in the Player control panel (see "Previewer Control Panel" on page 3). The computation is a conservative approximation of what may happen during playback: some problems shown may not occur during real playback, but if the bandwidth strip show no problems, your presentation is probably in good shape.

The following colors are used in the three representations of bandwidth:

Green No problems. The bandwidth is adequate to fit the media objects through the pipeline.

Amber Could be a problem. Amber color in the main part of the presentation denotes a potential stall. Amber used in the very beginning (before the zero second marker of the timeline) is the preload time. This does not indicate a stall, but the amount of time required to download the initial part of the presentation. An amber color in the middle of a presentation indicates a stall which coincides with a wait for user interaction — this is typically not a performance problem.

Red Definitely a problem. Red indicates that in most likelihood there will be a stall. When stalls occur, the streaming player goes into a state of re-buffering which means the presentation stops and data is recollected before it continues to play.

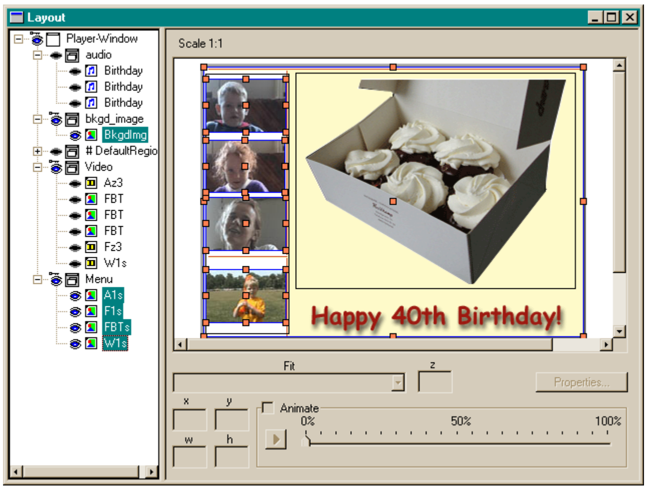
The GR*i*NS Previewer software does *not* have streaming capabilities enabled. The purpose of the bandwidth indicator is specifically designed for use with RealOne and HTML+Time exports.

Layout

Layout Overview

Where the structured timeline is used to create temporal behavior in a presentation, the layout view is used to create the spatial behavior. This is done by creating *regions* which are rectangular planes to which data is rendered.

Each region is created with a series of co-ordinates, a Z-index and a fit behavior that determines how media fill the presentation canvas. The follow view illustrates a generic Layout view structure:

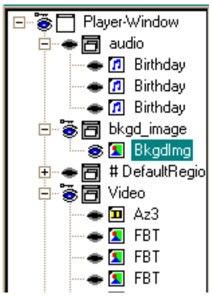


The Layout view is divided into three parts: the *browser*, *layout* and *property* panes.

Layout View Panes

Browser Pane

This is a hierarchical view (tree widget) containing the media objects in the presentation and their region assignment. The browser pane has full drag/drop functionality. This view is especially useful for moving media between regions. It also has a number of informative icons to help with visual arrangement:



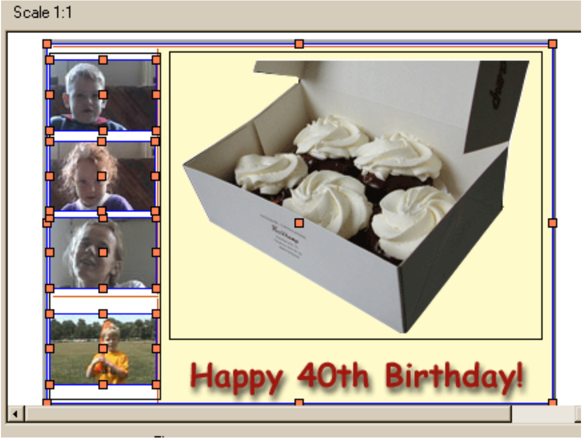
Top-layout The outer document window. All of the regions and media assets in the presentation are nested in this layout except HTML content in the Browser and Context windows..

Region The icon denoting a region. It will contain the media children.

Media Icons denoting the different media items. Each have an eye icon (it changes color when the object is selected for view) and a type icon. The name (if any) is also shown.

Visible/ Invisible key If you click on the eye icon, a lock key appears, to indicate that the associated item will be locked into view even if another object is selected in the tree. Click again to unlock. (Used when aligning media in different regions.)

Layout Pane



The layout pane is where all regions and their content are displayed. Most often, the region/media to be displayed will be selected in the Browser pane. Selecting a region in the Layout pane has the following semantic:

- if no other object is selected, then the object with the highest z-order under the mouse will be selected;
- if another object is already selected, clicking within that object retains its selection, and click outside the object, it follows the rule above.

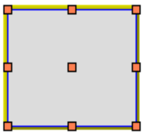
Using the ctl+click method will allow the selection of multiple regions, as in the browser pane.

When working with regions, you can manipulate either a region outline (resulting in changes being applied to all media objects using that region) or on a single media item selected via the browser pane (resulting in changes made only that media item).

Resizing a region will only resize the associated media items if the value of the *fit* attribute allows such resizing.

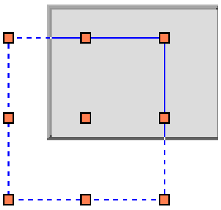
When working with regions, some functions can be achieved by using mouse clicking and dragging actions:

Region selection/ Resizing



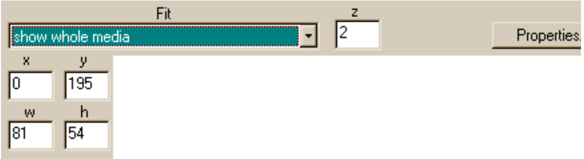
The currently selected region will show drag handles, allowing positioning and resizing of the region. Note that when working with hierarchical regions, repositioning a parent region will also reposition its children.

Region/media positioning



Content media within regions are clipped to their parent. Positioning can occur with both regions and media. Therefore, a media or region child object can be positioned outside of its parent region, rendering part of it invisible. This is denoted by drawing the child with a dotted line. (This is particularly useful for animations that appear to pan in from a direction.)

Property Pane


Fit: show whole media
z: 2
Properties
x: 0
y: 195
w: 81
h: 54

The property pane contains common properties for the currently selected region or media object. Right clicking an object in either the browser pane or layout pane will allow the complete property window to be shown. Less often used region attributes are accessible via the properties box.

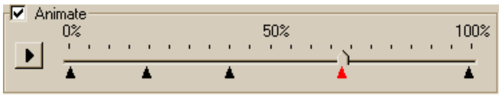
Fit The fit attribute determines the scaling of the media object within its region. In the case of regions, the fit attribute will specify the default fit for all child media objects unless overridden on the object.

z The z-index attribute specifies the top-to-bottom layering of regions and media. A higher z-index value means that the object is more to the front.

Properties Opens the main property window for the currently selected object. The property tabs and attributes are described later in this manual.

x/y/w/h Positioning coordinates that can be a pixel value. Positioning via percentages is possible via the geometry tab on the properties sheet (accessed via the property button). When using percentage-based placement, relationships among items are preserved if the containing window is resized in the RealOne player.

Animation


☑ Animate

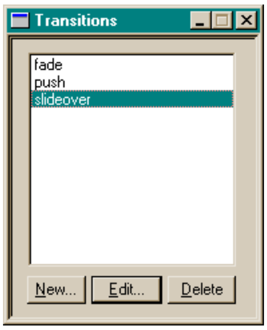
When a media object is selected, the animate checkbox becomes active, allowing for animations to be applied to that object's region.

The animateNode functionality allows a media node or its containing region to be moved and resized over time. When a media object has the animate checkbox enabled, multiple key frames can be defined along a motions path. To use animation effectively:

1. Enable the animate check for the currently selected object.
2. The red arrow indicates the currently selected keyframe.
3. If the media object in the layout pane is currently positioned in the desired *starting* position, drag the progress bar to the far right so it is positioned at 100%. This will specify that the end animation position will be recorded. Also note that the last keyframe becomes red.
4. When the 100% keyframe indicator is illuminated red, drag the media object to its desired destination.
5. Press the play on the right of the progress meter to watch the animation in action.
6. Should additional key frames are to be added, stop the playback and position the progress bar to an new point somewhere between and end.
7. Position or size the media object in the layout pane and note that a new red keyframe appears.
8. Playback the animation again and note that additional movements have been added.

Transitions View

Transitions Overview



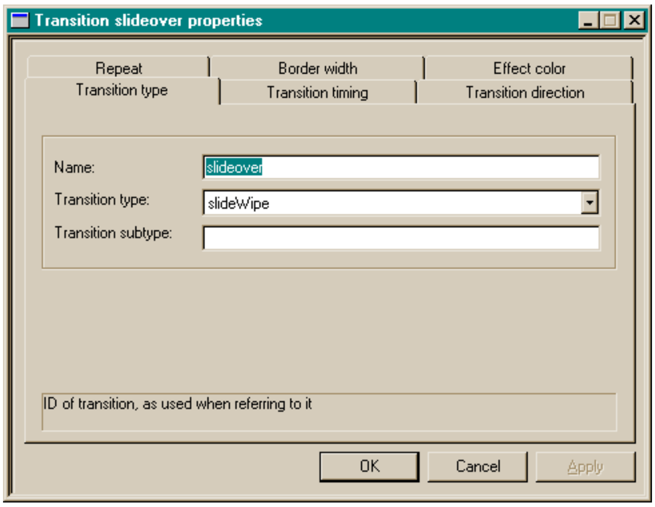
fade
push
slideover

The *Transition view* shows the set of transitions that have been defined for the current presentation. Each of the GR*i*NS design templates contain a set of transition, which may be extended by a presentation author.

Each transition has a standard set of properties, such as type, sub-type, duration, and direction. The properties can be manipulated via the transition view's Edit button. Once a transition is defined, it can be applied to a media object as an *input* or *output* transition. When relevant, an object's property box will contain a tab called *Transitions* where the transition used for the begin and end of that object can be defined.

Transitions Definition Options

The set of properties associated with a transition are illustrated in the following figure:



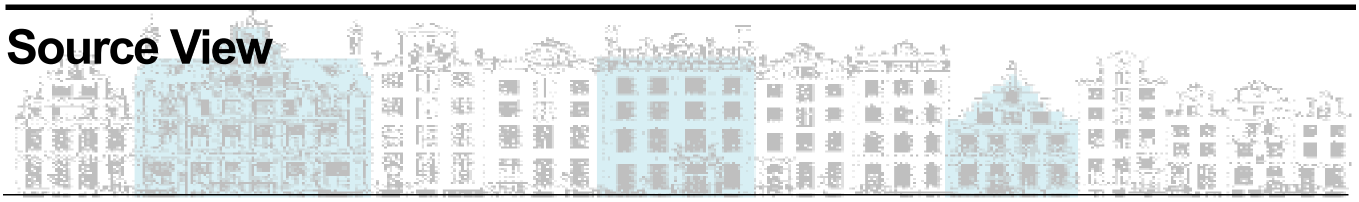
Name: slideover
Transition type: slideWipe
Transition subtype:

ID of transition, as used when referring to it

NOTE: While the transitions architecture has been standardized in SMIL 2.0, the supported set of transition sub-types has not. All SMIL players are required to support at least one of each of the four primary SMTE transition classes, but most players do not support the hundreds of sub-types. Consult the RealOne and IE-6 documentation for the current set of supported transition classes.

See "Transition Properties Tabs" on page 49 for the description of each property tab.

Source View



Source Viewing and Editing

Overview

Since SMIL is an XML-based language, the actual creation of presentations can be done in a text editor. Of course, the power of GR*i*NS is that all of this work is done visually, with embedded preview and bandwidth monitoring. (This enables authors to spend a fraction of the time otherwise required with a text editor.)

While a visual interface is useful in most cases, there are times when viewing the SMIL source generated by the editor also can be useful — such as during debugging or when learning the language. Therefore GR*i*NS also has a source view that allows the author to view what the actual end result will be.

The functionality provided by the source view is differentiated by product.

GR*i*NS/ RealOne A plain source view. It has no editing capabilities and merely displays the source.

GR*i*NS/Pro This source view includes full source editing capabilities as well as search and replace features.

Editing Commands (GR*i*NS/Pro Only)

This section denotes associated commands for the GR*i*NS/Pro editor product only.

Apply When doing manual edits to the source, the changes must be applied before the document is affected. The reason for this is that GR*i*NS' internal parser provides error checking that a normal text editor does not. Should errors occur, the author is notified by a dialog box.

Revert When editing the source, should the author want to revert back to previously before making the changes, this command-button will erase the changes. However, this does not work if the changes were applied.

The following commands apply to source editing:

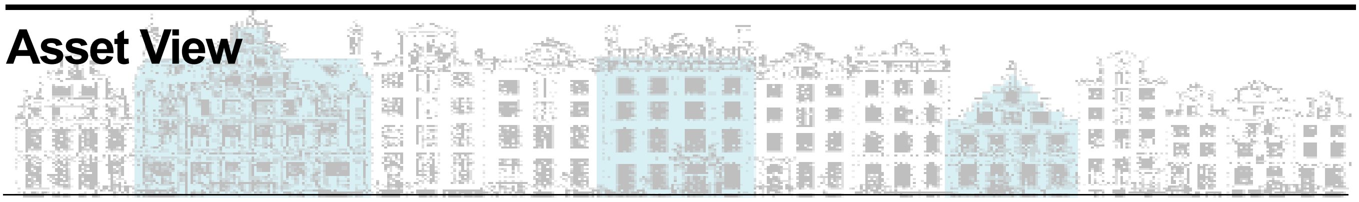
Find See "Find..." on page 9

Find next See "Find next" on page 9

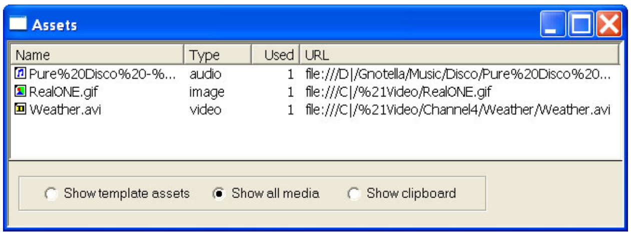
Replace See "Replace..." on page 9

Select node from source See "Select node from source" on page 13

Asset View



Assets Overview



Name	Type	Used	URL
Pure%20Disco%20-%...	audio	1	file:///D|/Gnotella/Music/Disco/Pure%20Disco%20...
RealONE.gif	image	1	file:///C|/%21Video/RealONE.gif
Weather.avi	video	1	file:///C|/%21Video/Channel4/Weather/Weather.avi

○ Show template assets ● Show all media ○ Show clipboard

When authoring large and complex presentations, it is usually important to be able to manage the media objects that are used in that presentation. GR*i*NS provides an Assets view, which holds media objects and presentation sub-structures; these may be integrated into a template or a document.

In GR*i*NS/Pro, you may create composite asset templates containing the structure for a (potentially) complex sub-structure, such as a news story. During editing, this structure can be imported into the document and filled with media content by the document author. Several standard GR*i*NS templates make use of this facility.

Asset View Options

The asset view has three options for organizing various kinds of media used during a presentation and identifying where where they are kept. A radio widget is supplied at the bottom of the window to change organizational view.

Show template assets Show the media/structure items that are available for re-use in the presentation. These fragments are stored locally in the .grins file. You can add new assets via drag and drop from the structured timeline view, or by dropping items from your system directly into the asset view window.

Show all media A full listing of all media used in the document. This view is read-only.

Show clipboard Showing contents of the clipboard. Structure can also be kept here. This view is read-only.

Each of the organizational views has up to four columns:

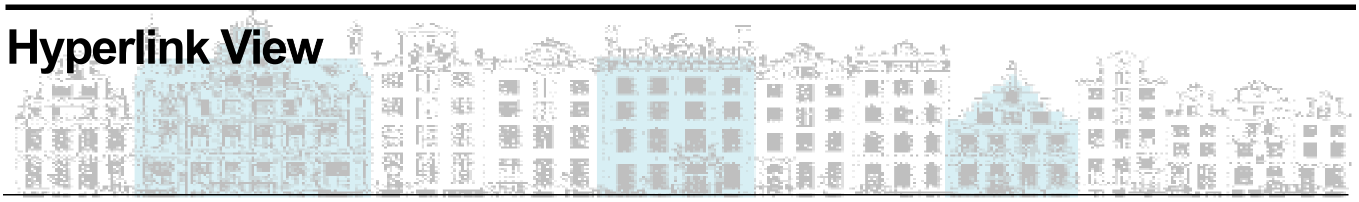
Name The name of the object.

Type The media type of the file

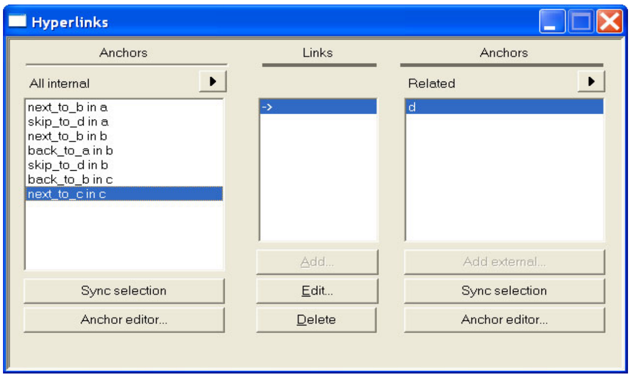
Used Whether or not the media object has been used in the presentation and how often.

URL The fully qualified pathname of the object on the local filesystem. This can also be URLs to the source objects.

Hyperlink View



Linking Overview



next_to_b in a
skip_to_d in a
next_to_b in b
back_to_a in b
skip_to_d in b
back_to_b in c
next_to_c in c

->

d

Hyperlinks provide a facility to associate one part of a presentation with an internal or external link to another (part of) a document.

Unlike HTML links, SMIL-based links also allow the activation and termination of links to be automatic (self-firing) and to be constrained by temporal attributes. The RealOne player also provides two HTML windows (the *context* and *browser* windows), which can contain Web content synchronized to the SMIL player content.

GR*i*NS provides a comprehensive management interface to maintain links in the presentation.

Link Management Commands

Hyperlinks source and destination can be visually created in the structured timeline, but they cannot be edited from that view. Modification and deletion of hyperlinks can only be done in the Hyperlink view. The view is divided into three sections: source anchor, links and destination anchors.

Anchor source The left pane of the view identifies the source of the hyperlink. By default when creating a hyperlink, they are unnamed. But in this view, the anchors can be given names. Generally the anchor name should reflect the action that is desired by the hyperlink, rather than the name or the source or destination.

Links The center section of the view that denotes a connection between source and destination. This link can be any of three directions, but that the most general usage is forward.

Anchor destination The right pane of the view that shows the destination of the link. Both anchor panes do not actually show if a connection is present, but show a list of qualified anchors that may or not be connected.

The anchor destination pane also comes equipped with a toggle button that will filter the types of anchors by their used relation:

All internal A full listing of all internal anchors, whether used or unused. Should a connected source and destination be selected in the appropriate panes, the Links pane will reflect that the connection exists.

Dangling Show anchors that have no hyperlinks attached to them yet.

Follow global focus Show anchors on the objects currently selected in the structure view.

All related anchors This list shows only the existing connection to the anchor that is selected in the anchor source pane. If there are no connections from the source, this pane is empty.

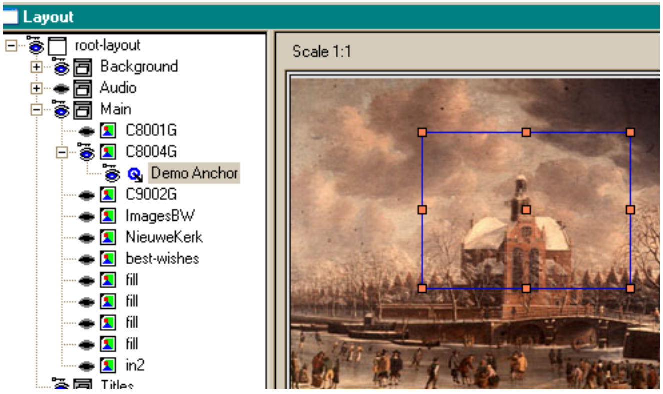
No anchors, links only Makes the right pane empty, but shows all links that originate from the anchor that is selected in the left pane.

External Similar behavior to the *All related anchors* option except that it relates to external links only. This is used to create links to external documents such as other presentations or web-pages.

Keep list Freezes the current contents of the pane.

Creating Anchors on Media

Anchors on media are created in the Layout view.



Links in SMIL can be associated with whole nodes or with 'hot-spots' that cover only part of a node. To create partial node anchors, open the layout view and select an object from the browser pane's selection tree. Next, perform the following steps:

1. Select Linking->Create Full Link Source and Edit...
2. "Open" the image in the layout selector, and click on the anchor symbol.
3. Using the drag handles in the layout view, create an anchor shape.

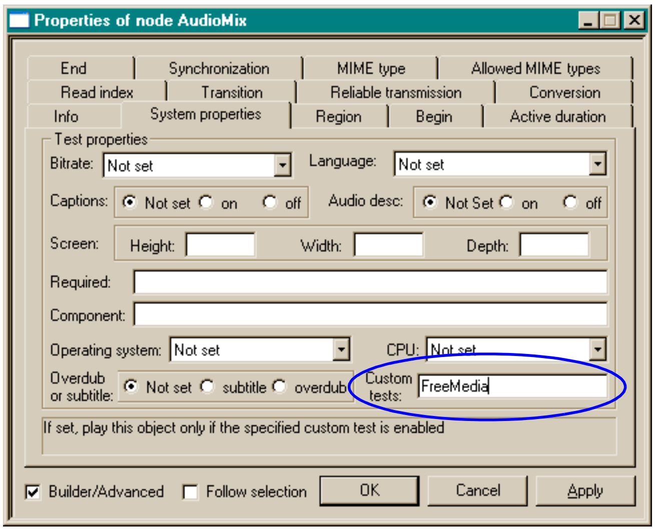
(You may also set anchor coordinates by hand under the More Anchor Setting property tab. See "Anchor Property Settings" on page 55.)

Custom Tests (GR*i*NS/Pro only)

Custom Test Overview

Custom tests allow users to create their own document test parameters. Unlike system test parameters in SMIL, custom tests can be designed to meet the need of the application, rather than the characteristics of the environment.

Associating a Custom Test



Properties of node AudioMix

Bitrate: Not set

Language: Not set

Captions: ● Not set ○ on ○ off

Audio desc: ● Not Set ○ on ○ off

Screen: Height: Width: Depth:

Required:

Component:

Operating system: Not set

CPU: Not set

Overdub or subtitle: ● Not set ○ subtitle ○ overdub

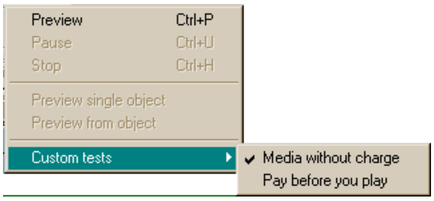
Custom tests: FreeMedia

If set, play this object only if the specified custom test is enabled

☑ Builder/Advanced □ Follow selection

A defined custom test variable can be associated with an object via the Properties->System properties tab. Enter the name of the desired custom test attribute in the space provided. If the value of the custom test attribute is True when the object is (to be) activated, it will be rendered.

Previewing a Custom Test

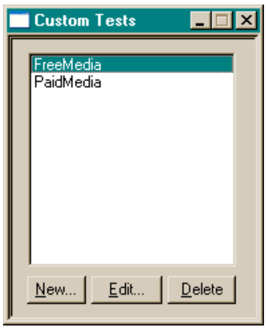


A custom test that has been defined to be VISIBLE may have its state changed during previewing via the Preview->Custom test menu.

To set a custom test to TRUE, select the appropriate label.

Note that SMIL does not support mutually exclusive custom tests automatically.

Manage Custom Tests



FreeMedia
PaidMedia

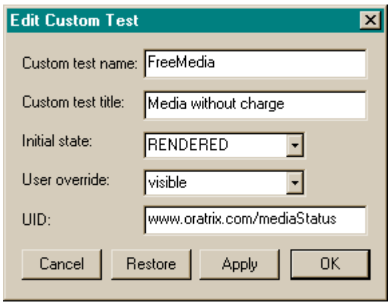
The set of custom tests for a document may be managed via the Windows->CustomTest menu. The options available are:

New ... Create a new custom test variable

Edit ... Change the settings for an existing (highlighted) custom test variable.

Delete Remove the custom test variable.

Create/Edit Custom Tests



Custom test name: FreeMedia
Custom test title: Media without charge
Initial state: RENDERED
User override: visible
UID: www.oratrix.com/mediaStatus

The Edit Custom Test property box is available from the Window->CustomTest menu. It has the following properties:

Custom Test Name The name given to the CustomTest attribute

Custom Test Title A text string used as a label in the Preview->CustomTest menu selector.

Initial State: One of the following values:
RENDERED: evaluate the test to TRUE
NOT RENDERED: evaluate the test to FALSE

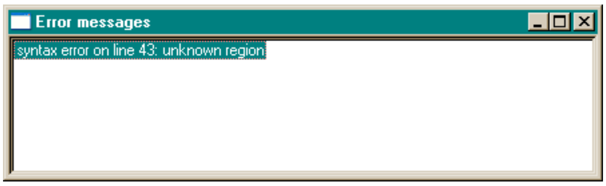
User Override Does the user have the ability to override the custom test at run-time (assuming the user agent can support setting custom test values):
VISIBLE: allow the user to override
HIDDEN: do not allow the user to override

UID Identifier than can be used to set the value of the custom test.

Source File Error Messages

Message Management (GR*i*NS/Pro)

The Error Messages view provides an interactive interface to finding and fixing errors in SMIL source files. A scan of the source file is made at parse time (when the document is loaded). If any errors are found, they are listed in an error window managed by the Error Messages view. If you select an error in this view, the source editor is activated so that you can correct the problem. (You may also select automatic correction.)



syntax error on line 43: unknown region

As this view is closely coupled to the GR*i*NS Source Editor, it is only available for the GR*i*NS/Pro product. Users of GR*i*NS/RealOne are given the option of accepting automatic error corrections for problems found during document parsing.

Property Dialogs

Overview

Property dialogs contain all attributes that control the behavior of an object in the presentation. Property dialogs contain a set of tabs, arranged in a series of classes.

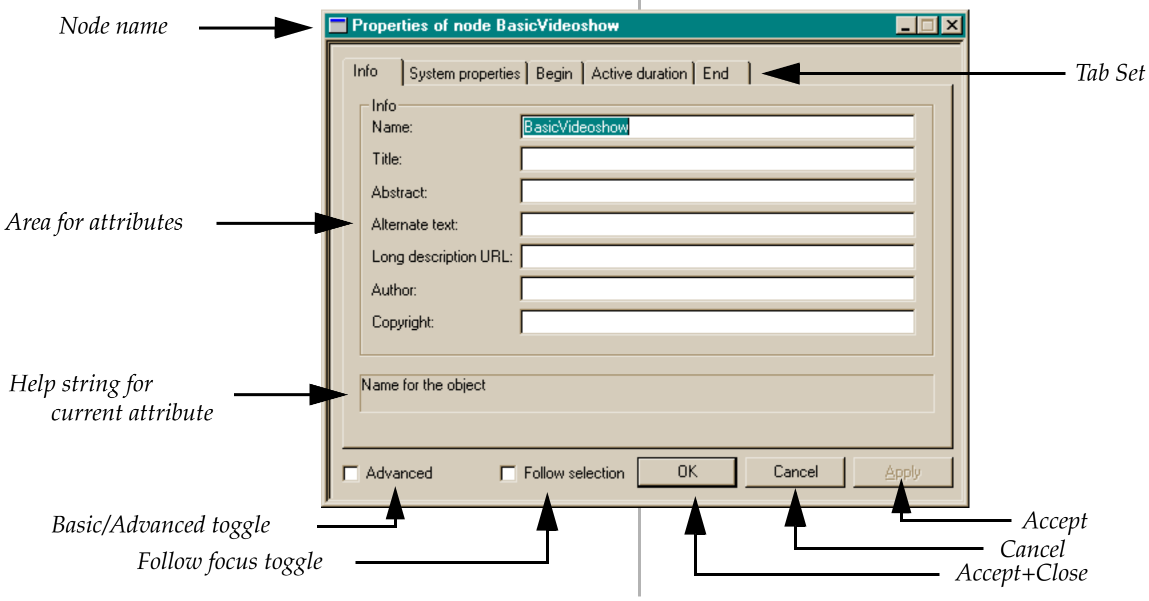
All property dialogs are contextually sensitive to the object that is currently selected; as a result, not all property dialogs will always contain the same set of tabs. In addition, the tabs available in a property box are grouped into *Basic* and *Advanced* sets. (For common template-based editing, most users can turn off the Advanced setting.)

The general architecture of the property dialog is shown below:

A good understanding of the elements, attributes and legal attribute values in SMIL 2.0 is highly recommended in order to gain full benefit from the property dialogs in GR*i*NS.

In this chapter, a comprehensive list of the available tabs will be explained, although not necessarily in the order of appearance in the consecutive tabs.

In each of the sections below, the various attributes on a particular tab are illustrated and described. In some cases, the number of attributes available on a tab will depend on the setting of the Advanced toggle. In these cases, both the Basic and Advanced mode will be illustrated. (In GR*i*NS/Pro, the Advanced mode button is labelled *Builder/Advanced*.)



Properties of node BasicVideoshow

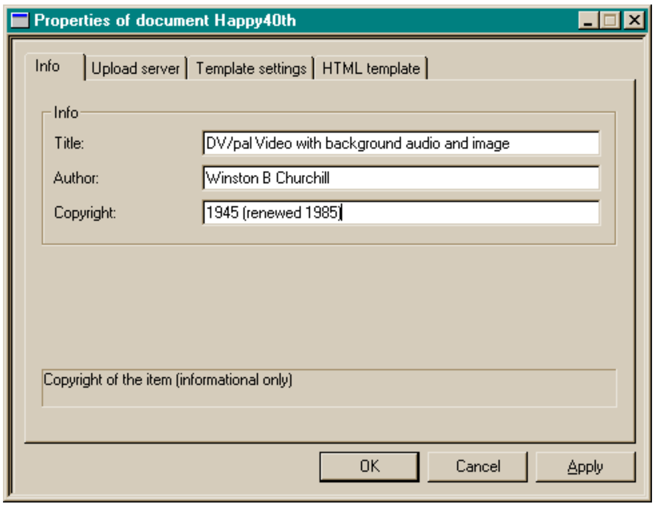
Info
Name: BasicVideoshow
Title:
Abstract:
Alternate text:
Long description URL:
Author:
Copyright:

Name for the object

Document Properties

The Document Properties dialog provides copyright and author information, and allows the (experienced) user to set information used during presentation upload to a media server. (See “Publish for RealOne and upload ...” on page 8.) Many of the values for the properties must be obtained from the system administrator of your media server.

Info



Info

Title: DV/pal Video with background audio and image

Author: Winston B Churchill

Copyright: 1945 (renewed 1985)

Copyright of the item (informational only)

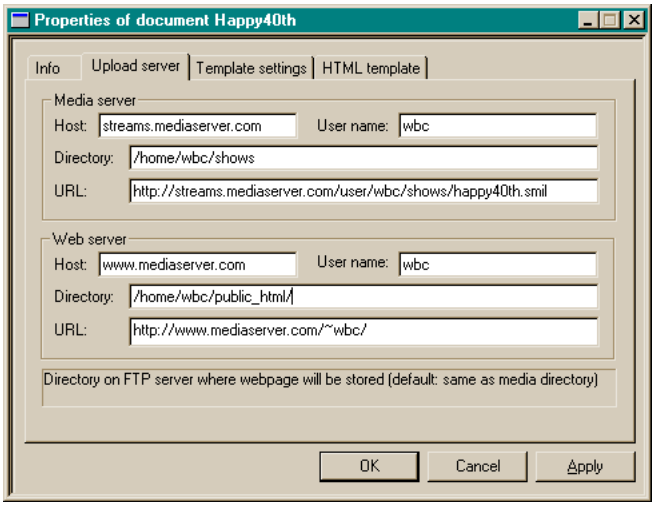
This tab contains general information on the presentation and is placed in the published *.smil* document.

Title The name of the presentation. The default value is taken from the template used.

Author The name of the author. The default value is taken from the template used.

Copyright The copyright date.

Upload Server



Media server

Host: streams.mediaserver.com User name: wbc

Directory: /home/wbc/shows

URL: http://streams.mediaserver.com/user/wbc/shows/happy40th.smil

Web server

Host: www.mediaserver.com User name: wbc

Directory: /home/wbc/public_html/

URL: http://www.mediaserver.com/~wbc/

Directory on FTP server where webpage will be stored (default: same as media directory)

This tab contains information used during a “publish with upload” request. (See: “Publish for RealOne and upload ...” on page 8.)

Media Server settings:

Host The (ftp) name of the Real server for the presentation.

User name The (ftp) login name for the user. (A password promt will be requested.)

Directory The (ftp) directory where the .smil file will be put. (A sub-directory will be created for the data.)

URL The URL of the .smil document on the server. (Note a .smil file name is required; it is usually the same as the name used during publishing.)

Web Server settings:

Host The (ftp) name of the Web server for the presentation. (Defaults to the name of the media server.)

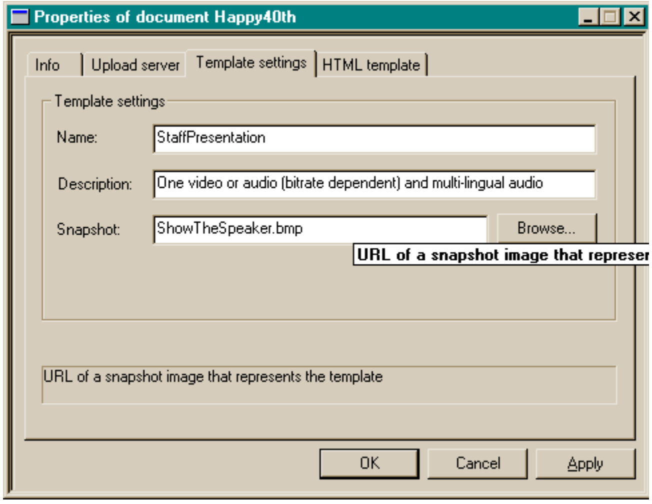
User name The (ftp) login name for the user. (Defaults to the user name for the media server; a password may be requested.)

Directory The (ftp) directory where the .ram file, the HTML page generated, and Web pages used in the RealOne *context* or *browser* windows will be put.

URL The URL of the directory containing the Web pages used in the RealOne *context* or *browser* windows as seen from the .smil file. This URL will be prepended to all RealOne *context* or *browser* window page references that are defined as local HTML pages during GRiNS editing.

Note: When a user associated content in a presentation with the RealOne context or browser windows, he may reference either a global HTML file (that is, one starting with http://...), or a local HTML file (that is, one consisting of a local name). During upload, local HTML pages are copied to the Web server, and the references in the .smil file generated by GR*i*NS are changed to reflect the new location.

Template Settings (GR*i*NS/Pro Only)



Template settings

Name: StaffPresentation

Description: One video or audio (bitrate dependent) and multi-lingual audio

Snapshot: ShowTheSpeaker.bmp

URL of a snapshot image that represer

URL of a snapshot image that represents the template

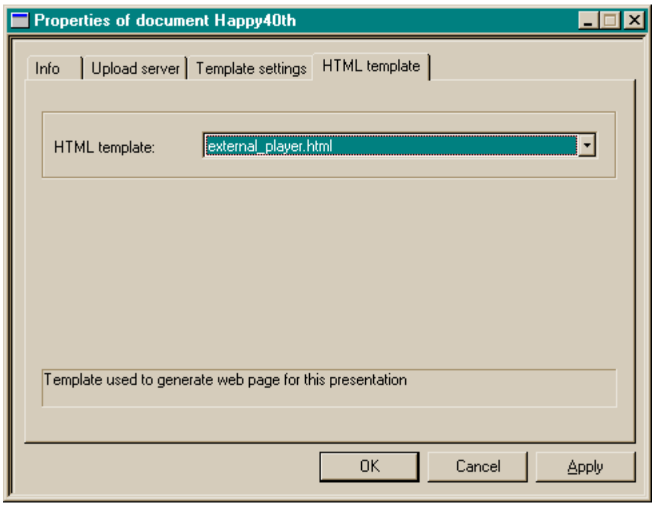
When Template Builder mode is selected (see: "Template builder" on page 41), this property box defines the basic attributes of a new template.

Name The name displayed in the template selection box at startup.

Description The descriptive text displayed in the template selection box at startup.

Snapshot The icon (in BMP format) displayed in the template selection box at startup.

HTML Template (GR*i*NS/Pro Only)


Properties of document Happy40th

HTML template: external_player.html

Template used to generate web page for this presentation

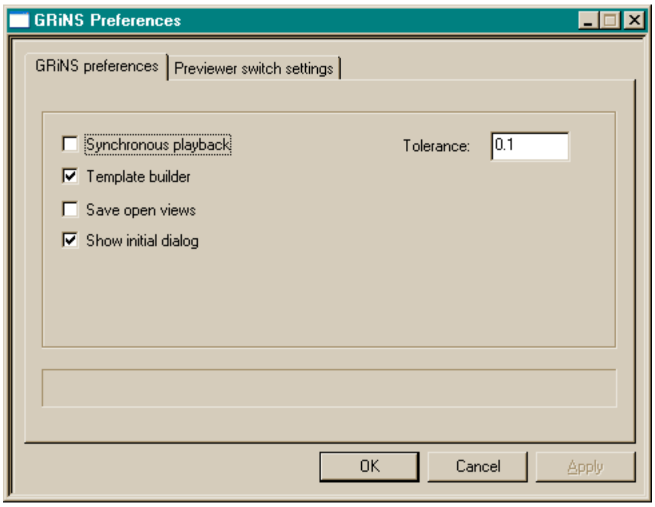
When Template Builder mode is selected (see: “Template builder” on page 41), this property box defines the name of an HTML template for generating a Web page announcing the presentation. A sample template is included with the distribution.

HTML template The name of an HTML template that is used to generate a Web page announcing your presentation. The default HTML template is defined in the presentation template.

Preferences (GR*i*NS/Pro Only)

GR*i*NS/Pro has a number of user preferences that are saved to the document that are specific to GR*i*NS. These special attributes can be modified by the File-Preferences pulldown menu.

GR*i*NS Preferences


GRiNS Preferences

☐ Synchronous playback

Tolerance: 0.1

☑ Template builder

☐ Save open views

☑ Show initial dialog

Synchronous playback Because the GR*i*NS player does not stream media such as the RealOne player, computation of events may not be timed precisely against the playback of natural duration media such as audio and video. The result is that manually timed events that are synchronized to cues in an audio or video media item can drift and appear to playback incorrectly. This check will cause the clock to stop the playback of media items in order to be 'frame accurate' to the events that were programmed. The side effect is that audio and video may be jerky, but it ensures that the export of the presentation to a commercial player will be accurate.

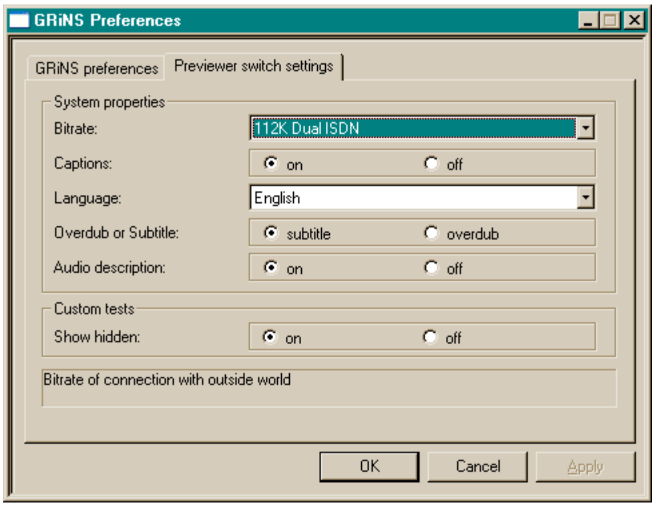
Tolerance The value of drift in seconds allowed when the synchronous playback check is enabled.

Template builder **(GR*i*NS/2 Pro)** This check enables many attributes to appear in the property dialog sets. They are attributes that would generally be needed by an author whose intention is to create templates for content authors.

Save open views Records the positioning of windows in the GR*i*NS workspace for user convenience.

Show initial dialog The default setting of the first pop-up dialog asking to open or create a project.

Previewer Switch Settings



GRiNS Preferences

System properties

Bitrate: 112K Dual ISDN

Captions: ● on ○ off

Language: English

Overdub or Subtitle: ● subtitle ○ overdub

Audio description: ● on ○ off

Custom tests

Show hidden: ● on ○ off

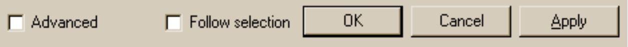
Bitrate of connection with outside world

This property box allows all of the system test variables to be set for the current project. Note that only objects that have test variables associated with them will be impacted by any of the settings.

These settings also determine how .smil files are generated during pruning and publishing. If Prune is selected from the Publish menu, only the items that correspond to the setting in this box will be inserted into the file.

General Property Tab Controls

Every property dialog contains a set of 2 checkboxes and a series of three command buttons.



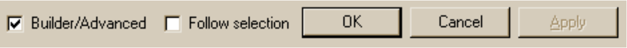
Advanced This checkbox provides access to advanced attributes and property tabs. For most common editing functions, this can be left unchecked (to Basic mode).

Follow selection This allows for the modal snapping of the property dialog to objects that are currently selected. The property dialog will follow focus to objects being selected in other views.

OK/Cancel Commit/cancel changes and close the property dialog.

Apply Commits changes leaving the property dialog open. Note: when using the follow selection option, changes should be applied before moving focus to the next object. If this is not done, the user is notified by a popup dialog if they want to commit the changes first.

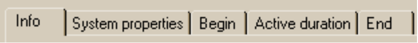
In GR*i*NS/Pro (only), when the template option has been selected from the Preferences dialog, the bottom checkboxes are shown as follows:



Builder/ Advanced If selected, extra properties are shown. These include all properties normally under Advanced, plus extra options for creating templates.

Media/Time Container Properties

All media and time container (that is, PAR, SEQ, EXCL) share the following set of property tabs:



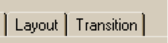
Info A tab containing general information on the node (such as name, copyright, etc.), plus (if a media node) the URL of the associated media object.

System Properties A tab containing the language and bitrate settings associated with the media or time container.

Begin / End Timing attributes that control when a node starts and ends.

Active Duration Timing attributes that control node duration, whether it repeats and where the object remains on the screen after the active duration expires.

In addition to the general properties, media containers also have the following property tabs available:



Subregion Layout properties of the media object as a child of its parent region.

Transition Attributes for setting input/output transitions.

In Advanced mode, media object also have the following tabs exposed:

Geometry Advanced sub-region positioning attributes

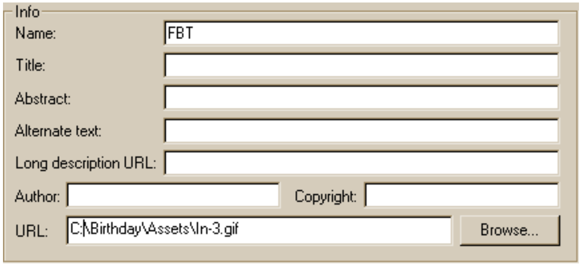
Conversion Attributes to control media conversion to RealMedia formats

Miscellaneous Control over attributes speciality attributes and attributes used on export to the RealOne player.

Common Container Tabs

Info Tab

Providing values of all Info attributes is optional.



Info

Name: FBT

Title:

Abstract:

Alternate text:

Long description URL:

Author:

Copyright:

URL: C:\Birthday\Assets\In-3.gif

Name The name given to the object. This is not required, but can be used for descriptive purposes. If events or links are used in a document, then names are automatically generated if not supplied by the user. If non-unique names are used (for instance, as a result of node copying and pasting), GR*i*NS will automatically make names unique on export.

Title Descriptive text.

Abstract Descriptive text.

Alternate text Similar to the ALT attribute in HTML.

Long description (URL) Pointer to extensive descriptive text.

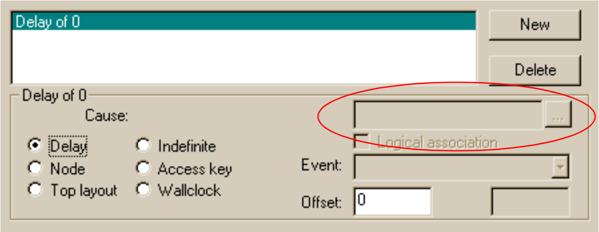
Author Informative dialog for the author's name of the object. (Not on all objects.)

Copyright Informative dialog for the copyright of the object. (Not on all objects.)

URL The physical location of the media object. Can be set to a Web URL, FTP site or a local objects. (Only local objects are copied upon publish.)

Begin/End Tabs

The begin and end attributes control the object's scheduled and event-based activation and termination. Since most of the attribute values are similar for begin and end, we describe them together in this section. Note that multiple begin events may be defined per object. The object will become active as soon as any of its begin conditions are satisfied.



Delay of 0

Delay of 0

Cause:

● Delay ○ Indefinite

○ Node ○ Access key

○ Top layout ○ Wallclock

☐ Logical association

Event:

Offset: 0

New This command button creates a begin or end event. The default delay is 0s; use *Offset* to define another value.

Delete This command deletes the selected begin/end condition.

Cause The class of begin/end offset:
Delay: a time value specifying seconds
Node: an object selection box appears; the *source event* type box becomes enabled.
TopLayout: An event may be associated with the opening and closing of the topLayout window(s).
Indefinite: a delay of indefinite specifies that the object remains active until its parent terminates
Access Key: indicate that a UI access key can be used to generate an event. (You must specify the key.)
Wallclock: the Entry field contains an absolute time.

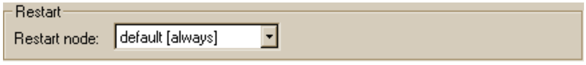
The circled area is a function-sensitive box that allows information relevant to the event type to be defined.

Logical association Creates an internal X-Pointer value; if the node containing this link is copied, the link is saved as a relative value.

Event The type of event that is the result of the Cause. This is a drop-down list of event types that can be applied to the object, such as firing when the user clicks or when the referenced node starts/stops.

Offset A numerical value that offsets the cause's event. Usually a time setting that is used as a delay time prior to the event firing. (Negative values result in a negative scheduling offset, but not in an earlier starting time for an object.)

Selecting the Advanced box will expose the following attributes (begin tab only):



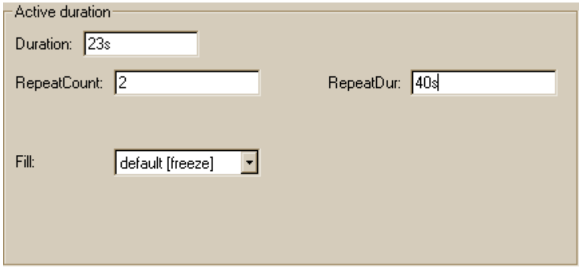
Restart

Restart node: default [always]

Restart node Determines whether a node should restart if a second begin event references it when its already playing.

Active Duration Tab

The Active Duration tab provides basic scheduling information for the node. (The values for this tab are often set by UI actions rather than direct editing.)



Active duration

Duration: 23s

RepeatCount: 2

RepeatDur: 40s

Fill: default [freeze]

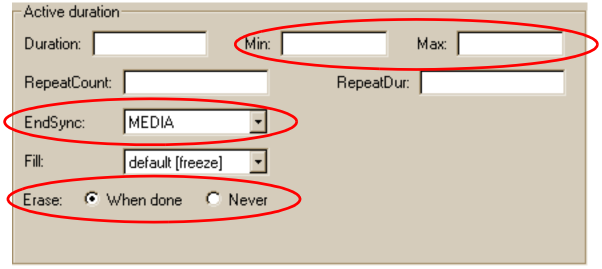
Duration An explicit time-value that overrides the object's defaut timing. If left blank, the object will either use the intrinsic value of the media or inherit the duration of its parent.

RepeatCount The number of times the object will repeat its duration before it is considered to have ended.

RepeatDur The total amount of time that the object should repeat for. This overrides the time calculated by *Duration*RepeatCount*.

Fill This attribute determines of a media object is left on the rendering surface after its active duration ends.

The following additional attributes are exposed in Advanced mode:


Active duration

Duration:

Min:

Max:

RepeatCount:

RepeatDur:

EndSync: MEDIA

Fill: default [freeze]

Erase: ● When done ○ Never

Min The minimum duration time of the object.

Max The maximum duration time.

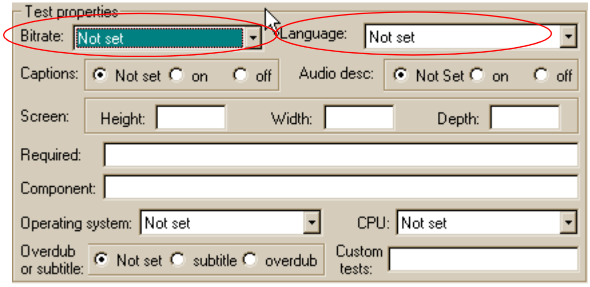
EndSync For use with parallel and exclusive container. This attribute specifies if the first, last or a named child should terminate the time container.

Erase An attribute that forces the object to be erased after its active duration has completed.

Additional Media Container Tabs

System Properties Tab

The System Properties tab contains test attribute settings for any node. The node is only played if the current player setting match the value of the attributes.


Test properties

Bitrate: Not set

Language: Not set

Captions: ● Not set ○ on ○ off

Audio desc: ● Not Set ○ on ○ off

Screen: Height: Width: Depth:

Required:

Component:

Operating system: Not set

CPU: Not set

Overdub or subtitle: ● Not set ○ subtitle ○ overdub

Custom tests:

Only the circled test attributes are available in all versions of GR*i*NS.

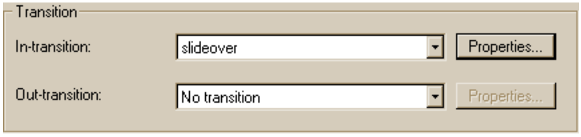
Bitrate The *bitrate* setting for the node and its children. Setting this will cause the object to skipped if the client’s connection is slower than the attribute value.

Language The language setting for the node and its children. Functions as a gate similar to *bitrate*.

In the GR*i*NS/Pro editor, all additional attributes (as defined in SMIL 2.0) may be set.

Transition Tab

The transitions tab allows an input and/or output transitions to be attached to a media object.



Transition

In-transition: slideover

Out-transition: No transition

These attributes are selectable as per the transition definitions in the transition view:

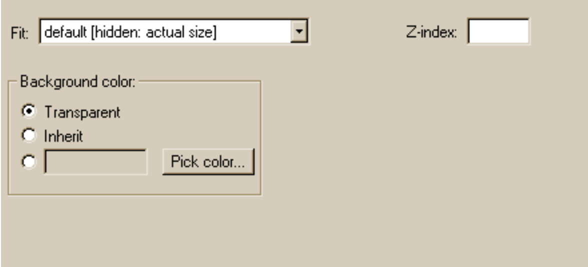
In-transition The transition to be used when the media object begins. The Property button to the right leads to the Transition property dialog (See “Transition Properties Tabs” on page 49).

Out-transition The transition to be used when the media object ends. The transition time is subtracted from the total duration of the object.

Subregion Tab

The Subregion tab provides access to the principal layout attributes for the object; these can be used to override the values for the associated region for rendering this object.

Most layout editing will occur via the Layout view and the associated properties. (See “Region Property Tabs” on page 50.)



Fit: default [hidden: actual size]

Z-index:

Background color:

● Transparent

○ Inherit

○

Should these be left blank or default, the properties of the region will apply to the object.

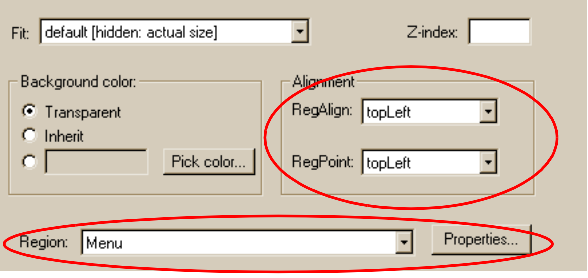
Fit The visual behavior of the media object as it is rendered to the region.

Z-index The top-to-bottom layering of the media object *relative to other objects in the same region.*

Background color A radio group that sets the color used as the background to the media object.

Important note: these values override the global region values only for this instance of this object. The actual region values are not effected.

The following attributes are exposed in Advanced mode:



Fit: default [hidden: actual size]

Z-index:

Background color:
● Transparent
○ Inherit
○

Alignment
RegAlign: topLeft
RegPoint: topLeft

Region: Menu

RegPoint A point on the media object that serves as the reference point for alignment.

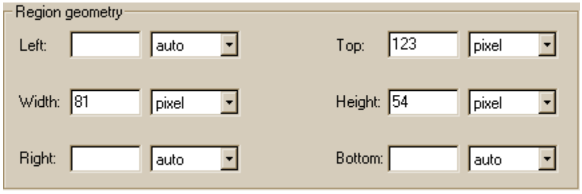
RegAlign The place within the region at which the object's RegPoint is placed.

Region The region in which the media object is rendered. The property button to the right leads to the Region property dialog (See "Region Property Tabs" on page 50).

Advanced Media Container Tabs

Geometry

The Geometry tab specifies sub-region placement attributes. These attributes allow media to be positioned within the current region; all attribute values are *relative to the containing region*.



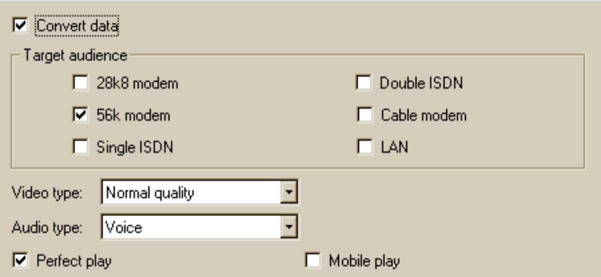
Region geometry

Left:		auto	Top:	123	pixel
Width:	81	pixel	Height:	54	pixel
Right:		auto	Bottom:		auto

Note that values can be specified as a pixel or percentage valye. By default, set to Auto; placement corresponds according to W3C CSS-2 absolute placement semantics.

Media Conversion

The conversion tab controls media conversion from native format to RealMedia. (Conversion only takes place if publishing to the RealOne player.)



☑ Convert data

Target audience

□ 28k8 modem
□ Double ISDN
☑ 56k modem
□ Cable modem
□ Single ISDN
□ LAN

Video type: Normal quality

Audio type: Voice

☑ Perfect play
□ Mobile play

Convert Data A checkbox that determines whether this media node is converted to RealMedia upon export.

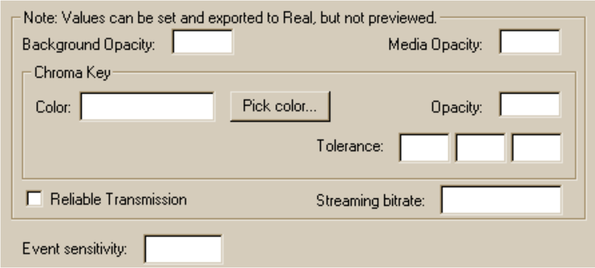
Target Audience A set of common bitrates that determine which RealProducer codecs are used to encode the media. If multiple values are selected, a RealNetworks SureStream file will be produced.
Note: SureStream technology is only recommended if a presentation is hosted on a Web site and not on a RealServer.

Video Type For video data, the type of image material. (Used to optimize encoding.)

Audio Type For audio data, the type of audio material. (Used to optimize encoding.)

Miscellaneous

These attributes cover a range of RealNetworks extensions to the SMIL language plus other speciality attribbutes. GR*i*NS will include these values in the published document, but the Previewer *may not* render the associated behavior.



Note: Values can be set and exported to Real, but not previewed.

Background Opacity:
Media Opacity:

Chroma Key

Color:
Pick color...
Opacity:
Tolerance:

□ Reliable Transmission
Streaming bitrate:

Event sensitivity:

Background opacity The percentage of opacity applied to a transparent background. (Valid for RealOne only.)

Media opacity The percentage of transparency applied to an opaque media object. (Valid for RealOne only.)

Chroma key color The color used for chroma keying. in which a particular color of a media object is rendered as transparent. (Valid for RealOne only.)

Chroma key opacity The opacity percentage for the chroma keyed color. (Valid for RealOne only.)

Tolerance The tolerance color value set in Red-Green-Blue allowing a differential to the chroma key color set to a media object. (Valid for RealOne only.)

Reliable transmission A directive to the streaming server to force reliable transmission even if this results in presentation delays. (Valid for RealOne only.)

Streaming bitrate The amount of bitrate that should be assigned to this object during streaming from a RealServer.

Event sensitivity An opacity percentage that determines if events are caught or passed to another media object.

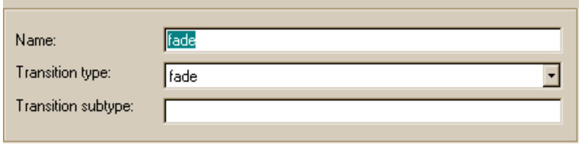
Transition Properties Tabs

Transitions are defined in the Transitions view. These table control the various properties associated with transitions.

Note that the GR*i*NS preview does not support all of the transition types, but allows all types to be defined and exported to the RealOne player.

Transition Type Tab

This tab allows a name, type and sub-type to be defined for a transition:



Name: fade
Transition type: fade
Transition subtype:

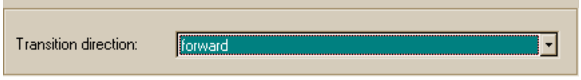
Name The name for the transition definition. Must be unique.

Transition type The major transition type used. The rendering of this effect depends entirely on the implementation of the player the document is exported to. The list as shown is fully supported by the GR*i*NS player.

Subtype A secondary transition type, defined by SMPTE. (Not all types are previewed.)

Transition Direction Tab

A single attribute to determine the direction of the transition, if applicable:

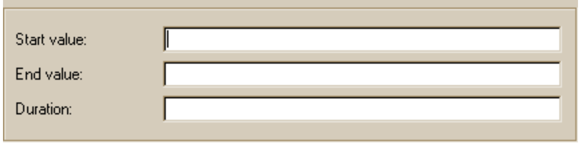


Transition direction: forward

Transition direction The playing direction of the transition, whether it be forward or reverse. Support depends entirely on the implementation of the target player.

Transition Timing Tab

This tab allows the duration of the transition to be defined:



Start value:
End value:
Duration:

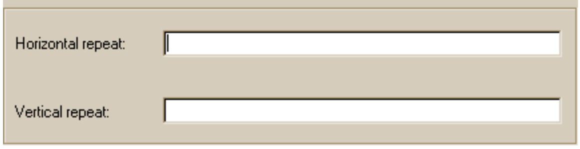
Start value Used to define where the transition should start as a percentage of the transition’s effect. (Default is 0.)

End value Used to define where the transition should start as a percentage of the transition’s effect. (Default is 1.)

Duration The timed duration of the transition definition. The default value is 1 second.

Repeat Tab

Two attributes are available on this tab to control the repeat behavior of the transition:



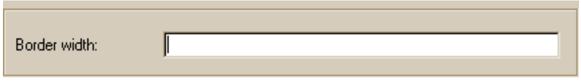
Horizontal repeat:
Vertical repeat:

Horizontal repeat The number of times that the transition will be repeated in its horizontal axis.

Vertical repeat The number of times that the transition will be repeated in its vertical axis.

Border Width Tab

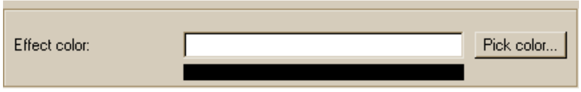
A single attribute to set the width of a Wipe transition border:



Border width This attribute applies only to the *wipe* transition. It controls the border size in pixels.

Effect Color Tab

This single-attribute tab is used to specify the color associated with certain transitions:



Effect color This attribute is used in conjunction with the *fade* transition and the *fadeToColor* or *fadeFromColor* subtype.

Region Property Tabs

When a region is created or edited in the layout view, the author is given access to two sets of tabs in Basic mode:

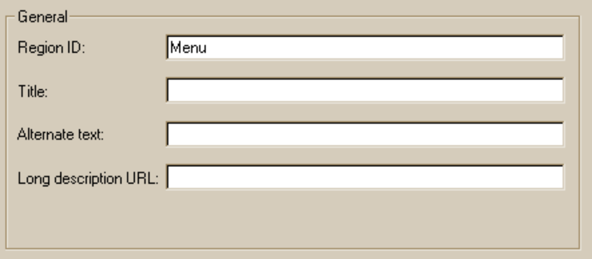
General A set of informational attributes on the region or topLayout, some of which are only visible in Advanced mode.

Layout Properties of the region, some of which are only visible in Advanced mode.

Basic Region Property Tabs

General Tab

The general tab for regions is similar to the Info tab for other objects.



General
Region ID: Menu
Title:
Alternate text:
Long description URL:

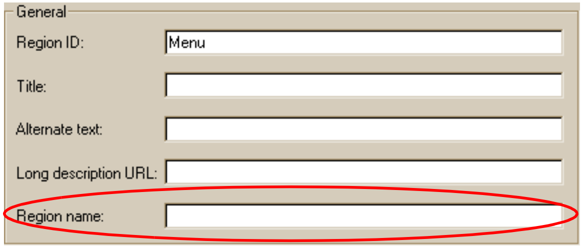
Region ID The referenced name of region that media objects will be assigned to in the presentation layout. If left as a null value, GR*i*NS will automatically generate a name.

Title An informative title that can be added to the region.

Alternate Text Alternative information that can be used to describe the region.

Long description URL A pointer to a text string that can be used to describe the purpose of the region.

The following additional attribute is exposed in Advanced mode:

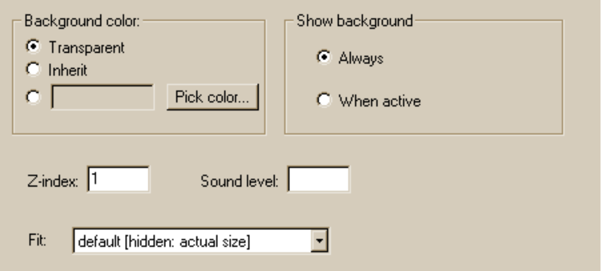


General
Region ID: Menu
Title:
Alternate text:
Long description URL:
Region name:

Region name A non-unique region name that is a synonym for the Region ID. Used in presentations that have multiple layouts embedded in a switch. The region is selected based on the name rather than the ID, since regions IDs must be unique.

Region Layout Tab

This tab provide basic information on the structure of the region:



Background color:
● Transparent
○ Inherit
○

Show background
● Always
○ When active

Z-index: 1

Sound level:

Fit: default [hidden: actual size]

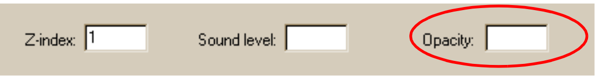
Background color The default background color assigned to the region. This behavior of this attribute depends on the value set for *Show Background*. (Only if SMIL Advanced Layout is supported.)

Show background Determines whether the background color of a region is rendered when the contents are *not* active.

Z-index The top-to-bottom layering of the region in relation to the master *Top-Layout* region. Higher values specify content that is more towards the front.

Sound level (For use with regions that contain audio objects.) A percentage value that controls the level of audio playback. Dafualt is 100%.

In Advanced mode, the following attribute is exposed:



Z-index: 1

Sound level:

Opacity:

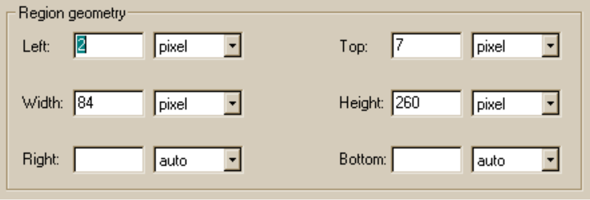
Opacity A percentage reduction of background color opacity. A default of 0% is equivalent to transparent.

Advanced Region Property Tabs

A single extra tab is exposed when in Advanced mode: the Geometry tab.

Region Geometry Tab

The Geometry tab allows basic region positioning to be defined.



Region geometry

Left:	2	pixel	Top:	7	pixel
Width:	84	pixel	Height:	260	pixel
Right:		auto	Bottom:		auto

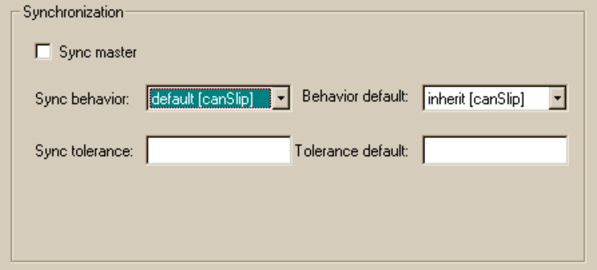
Note that values can be specified as being pixels or percentages. If percentages are used, the relative placement of objects will be preserved in the RealOne player when the view is magnified.

By default, values have the value of Auto, and placement corresponds according to W3C CSS-2 absolute placement semantics.

Builder/Advanced Media Container Tabs (GR*i*NS Pro)

Synchronization

The Synchronization tab contains attributes that affect the way media objects are scheduled and synchronized with respect to their parent.



Synchronization

□ Sync master

Sync behavior: default [canSlip] Behavior default: inherit [canSlip]

Sync tolerance: Tolerance default:

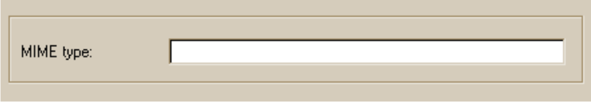
Sync master Enables the currently selected media object to become the synchronization master for its sibling elements.

Default The default sync master value for all children of the currently selected object or container.

Sync tolerance The tolerance value in seconds that an object is allowed to be out of sync from its master.

Default The default sync tolerance value for all children of the currently selected object or container.

MIME types



MIME type:

The MIME type of the currently selected media object can be overridden by this field.

Read Index

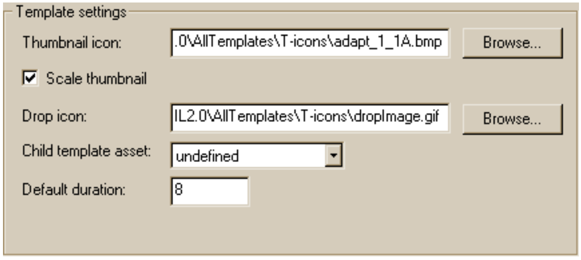
Read index:

When a presentation is viewed using special devices for disabled viewers, this field dictates the position in the media at which the descriptive text is read.

Template Options (GR*i*NS/Pro Only)

Three tabs are available to define default UI behavior associated with templates.

Per-Object Template Setting



Template settings

Thumbnail icon: .0\AllTemplates\T-icons\adapt_1_1A.bmp

☑ Scale thumbnail

Drop icon: IL2.0\AllTemplates\T-icons\dropImage.gif

Child template asset: undefined

Default duration: 8

Thumbnail Icon The icon (in .tif/.tiff format) displayed on a closed structure container.

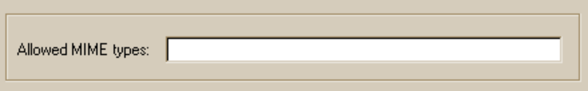
Scale Thumbnail Set if the thumbnail should be scaled along with the structure view.

Drop Icon The icon used to indicate when new media items can be dropped

Child template asset Select an asset type for constrained content. (Unused.)

Default duration The default duration of all children of this object.

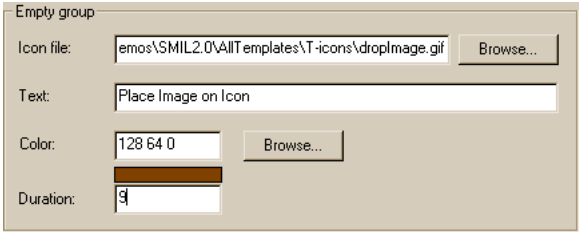
Allowed MIME types



Allowed MIME types:

Template builders may wish to create blank media objects that allow a specific MIME type to the object placed there. This option is an override to the server response for the media object.

Empty Settings



Empty group

Icon file: emos\SMIL2.0\AllTemplates\T-icons\dropImage.gif

Text: Place Image on Icon

Color: 128 64 0

Duration: 9

The Empty Group setting specifies how an object can be drawn when its structure container is open, but when it is not filled with any media. The attributes are:

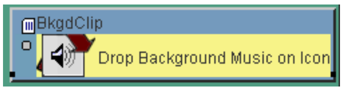
Icon file The name of an image that can be used to identify the type of the media to be placed in the container.

Text A line of text that is displayed when the container is open, but empty

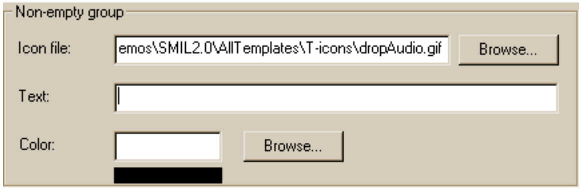
Color The background color of the text area.

Duration The (rendered!) duration when the object is closed. This does not impact the actual scheduled duration of the node, just the drawing space allocated for the object.

An example of an empty container is shown below:



Non-Empty Settings



Non-empty group

Icon file: emos\SMIL2.0\AllTemplates\T-icons\dropAudio.gif

Text:

Color:

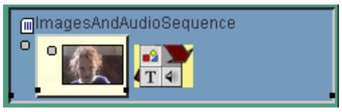
When a structure container in a template has content, a template author may still specify extra icons that invite users to add even more content. The attributes are:

Icon file The name of an image that can be used to identify the type of the media to be placed in the container.

Text A line of text that is displayed when the container is open and not empty

Color The background color of the text area.

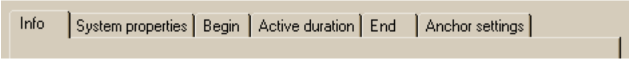
An example of a non-empty container is shown below:



Anchor Property Settings

Whenever a Link is created, the associated anchor has properties than can be set by the editor.

In Basic mode, the set of property tabs associated with Anchors is illustrated in the following image:



Info See "Info Tab" on page 43.

System properties See "System Properties Tab" on page 45.

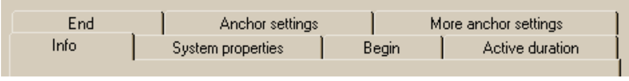
Begin See "Begin/End Tabs" on page 43.

Active duration See "Active Duration Tab" on page 44

End See "Begin/End Tabs" on page 43.

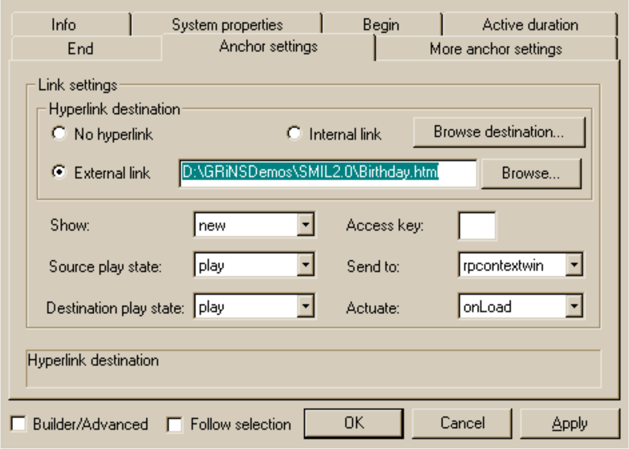
Anchor settings See "Anchor Settings Tab" on page 56

In Advanced/TemplateBuilder mode, the set of property tabs associated with Anchors is illustrated in the following image:



More anchor settings See "More Anchor Settings Tab" on page 57.

Anchor Settings Tab



○ No hyperlink ○ Internal link

● External link: D:\GRiNSDemos\SMIL2.0\Birthday.html

Show: new

Access key:

Source play state: play

Send to: rpcontextwin

Destination play state: play

Actuate: onLoad

□ Builder/Advanced □ Follow selection

This tab is used to set the properties of link anchors.

Hyperlink Destination properties:

No hyperlink Set on to effectively remove link.

Internal link A link to another node in the same (SMIL) presentation. (GR*i*NS/Pro only; not supported on export to HTML+TIME.)

External link The name or URL of a file. The target may be either an HTML file or a SMIL file. If a full URL is given, this URL is preserved on 'Publish'. If a local file name is give, the file is copied during 'Publish and Upload', and prepended with the URL specified in the Upload tab, as described in "Upload Server" on page 38.

Other properties:

Show Select the effect of following the link. Options:
New: create a new presentation, keep the old
Pause: create a new presentation, pause the old
Replace: create a new presentation, end the old

Source play state Determine what happens to the old presentation (if NEW was selected for Show):
Play: keep the source presentation going
Pause: pause the source presentation
Stop: end the source presentation

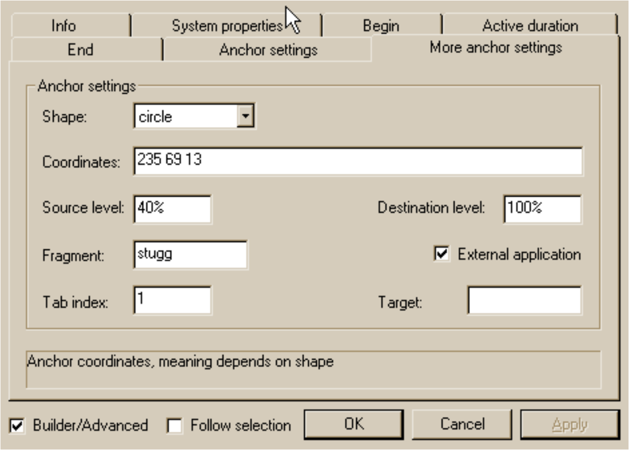
Destination play state The initial state of the destination of the link:
Play: keep the source presentation going
Pause: pause the source presentation.

Access key Defines the key used to actuate an anchor (default is the mouse key for Activate events).

Send to The name of the window in the presentation to receive the destination:
rpcontextwin: the context window in the RealOne player,
rpbrowserwin: the browser window in the RealOne player,
osdefaultbrowser: the media browser listed in the player's OS to handle that media type,
rpengine: the media player window in the RealOne player.

Actuate: Specifies when the link gets actuated:
onLoad: when the media object is schedule to run,
onRequest: when the object is selected via the access key or pointer device.

More Anchor Settings Tab



Anchor settings

Shape: circle

Coordinates: 235 69 13

Source level: 40% Destination level: 100%

Fragment: stugg ☑ External application

Tab index: 1 Target:

Anchor coordinates, meaning depends on shape

☑ Builder/Advanced ☐ Follow selection

This tab defines additional properties of the anchor.

Shape The basic shape of the anchor. Choices:
Rect: a rectangle (specify coordinates as x,y,x,y)
Circle: a circle (specify coordinates as x,y,r)
Poly: a collection of points (x,y,x,y,...,x,y)

Coordinates The points defining the anchor, in the formats above.

Source level The audio output level of the source after following the link (defined a percentage value)

Destination level The audio output level of the destination after following the link (defined a percentage value)

Fragment The fragment ID (within the destination)

Tab index A numeric index used to tab through the selectable objects.

Target The name of the top-level window that will receive the destination.

Background Information on SMIL 2.0

The W3C's SMIL format for encoding multimedia presentations for delivery over the Web is a little-known, but widely used standard. First released in mid-1998, SMIL has been installed on approximately 200,000,000 desktops world-wide, primarily because of its adoption in RealPlayer G2/7/8, Quicktime 4.1, and Internet Explorer 6. In August, 2001, the W3C released a significant update to SMIL: version 2.0.

Design Goals

The goals for SMIL 2.0's design can be partitioned across three broad categories:

- *extend the functionality of SMIL 1.0*: the designers of the first version of SMIL purposefully kept the language simple and relatively frills-free. SMIL 2.0 provides some desirable additions, including: support for increased interaction, enhanced timing semantics, extended content control and layout facilities, plus new animation and transitions primitives.
- *maintain a declarative, XML format*: while the integration of multimedia content is often the province of scripted or other programmed definition, SMIL 2.0 was developed to remain a fully declarative rather than a procedural format. In this way, a SMIL description doesn't define *how* a presentation is implemented, but rather *what* it is that the author wants — the implementation of the specification left up to the SMIL player. Also, in keeping with the first version of SMIL, SMIL 2.0 is fully XML compliant.[1]
- *introduce a module-based structure*: where SMIL 1.0 was a simple, single 29-page specification, SMIL 2.0 has been defined in terms of over 50 modules that can be partitioned into 13 functional groups, described in over 500 pages. By using a module structure, key aspects of SMIL 2.0 can be integrated into other XML-based languages without requiring support for the entire SMIL 2.0 specification. Even before the release of the SMIL 2.0 Language specification, the modularization policy bore fruit: parts of SMIL have been integrated into several other XML languages — such as SVG — and more examples of module reuse are expected.

As of the summer of 2001, several commercial versions of SMIL 2.0 were available (the first of which, Oratrix's GR*i*NS/SMIL 2.0 player, was released in September, 2000), and major mass-market media companies such as RealNetworks and Microsoft had announced support for all or some of SMIL 2.0's features. The 3GPP consortium, the standardization body coordinating the deployment for next-generation wireless devices, has

[1] SMIL 1.0 was not only an XML-compliant language, it was the *first* XML language released by the W3C when it was published in 1998.

also recommended that SMIL 2.0 be used as the basis for wireless multimedia devices.

SMIL: what it isn't (and what it is!)

Often, multimedia presentations are characterized by their content, rather than their composition. In this sense, there has been some confusion as to what a SMIL presentation is, and what it isn't.

To begin with the latter, it is fairly easy to summarize what SMIL 2.0 is *not*:

- *SMIL 2.0 is not a Flash substitute*: Flash is a proprietary content media type that is primarily used for small animations. SMIL 2.0 is not a content media type, in the sense that it does not define any particular type of media (such as vector or raster images, videos, text or audio data). Instead of media content, SMIL defines XML-based media composition. A SMIL presentation can include Flash objects.
- *SMIL 2.0 is not an MPEG-4 substitute*: MPEG-4 is a highly touted (but lightly implemented and deployed) format for describing the contents and interactions of media objects. More precisely, MPEG-4 is a family of protocols that covers a wide range of media-related concerns, but not a specific solution to any one class of media presentation. As we will see, there has been considerable work done to coordinate the development of SMIL 2.0 and MPEG-4 through the XMT specification.
- *SMIL 2.0 is not a D-HTML substitute*: Dynamic HTML (D-HTML) was introduced as a way of introducing local time- and animation effects into a static HTML web page. While some of the animation primitives in SMIL 2.0 resemble the functionality of some uses of D-HTML, SMIL's scope is much broader than the local nature of D-HTML.

What, then, is SMIL 2.0? It is a collection of XML elements and attributes that can be used to describe the temporal and spatial coordination of one or more media objects. SMIL 2.0 allows a user to define how independent media objects are to be integrated during the lifetime of a presentation. That presentation may be delivered via a streaming server, or it may be played locally.

SMIL 2.0 defines 10 major functional groupings of elements and attributes. (See Figure 1.) Of these, *Timing and Synchronization* is the core of the SMIL specification. The functional groupings represent collections of individual SMIL 2.0 modules, each of which defines elements and attributes intended to address specific multimedia issues in a reusable manner. The number of modules per functional grouping varies from 2 to about 20. In general, the more modules per grouping, the finer the granularity of each module.

The modularization of SMIL 2.0 was intended to not only facilitate the reuse of SMIL functionality in other XML languages, but to also help define a number of SMIL *profiles*, each of which would provide a collection of modules that could be customized to the primary goal of the profile.

Figure 2 shows the initial partitioning of SMIL 2.0 profiles. Of these, the SMIL 2.0 Language and Basic

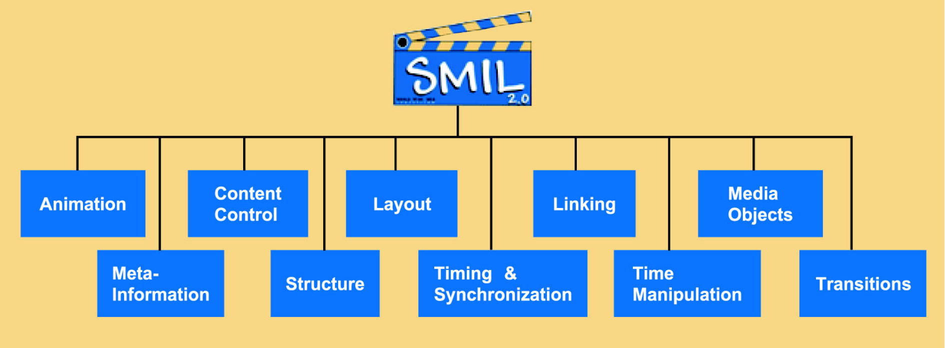


Figure 1: *SMIL 2.0 Functional grouping of module sets.*

profiles completed the required review and implementation requirements of the W3C for the summer 2001 release of SMIL 2.0. The XHTML+SMIL profile, which seeks to integrate SMIL timing, animation and transitions modules (among others) into XHTML, was not completed and remains under development by the W3C. In addition to the three SMIL profiles, a version of SMIL Animation was also released by W3C in July 2001 for final review by W3C members.

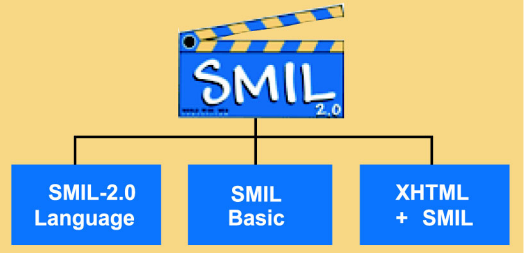


Figure 2: *SMIL 2.0 Profiles.*

Structure, Timelines and SMIL

A SMIL presentation is a *structured* composition of autonomous media objects. As illustrated in Figure 3, there are three basic timing containers that can be used in a SMIL presentation:

- *seq — sequential time container*: the children of a *seq* container are rendered in such a way, that a successor child never can begin before its predecessor child completes. A successor child may have an additional start delay, but this delay cannot resolve to be negative in relation to the end time of its predecessor.
- *par — parallel time container*: the children of a *par* container are all rendered in “parallel”. In terms of SMIL’s timing model, this does not mean that they get rendered at the same time, but rather that they share a common timebase defined by the *par* container. It also means that any or all of the children can be active at any time that the parent *par* is active. The *par* is the most general SMIL time container.
- *excl — exclusive time container*: only one of the children of an *excl* can be active at a time. The order of activation of the children depends on the *begin*

attribute of the *excl*'s children. Typically, each will have an event-based start condition such as:

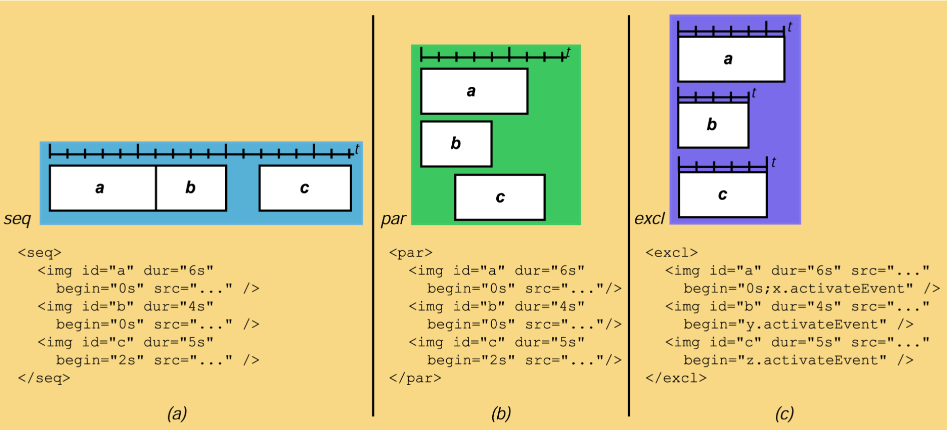
```
begin="button1.activateEvent"
```

that allows one of the children to start on demand. The *excl* container is new in SMIL 2.0.

The three basic time container types can be nested in a presentation hierarchy. That is, any child of either a *par*, *seq* or *excl* can be a simple media object or an embedded *par*, *seq* or *excl* container. As in SMIL 1.0, the hierarchy can represent relatively static presentation timing. The introduction of event-based activation/termination in SMIL 2.0 also allows a dynamic activation path to be defined. Note that most other multimedia formats only support a *par*-like semantic, not the *seq* or *excl* semantics.

Timeline Definition

A fundamental property of multimedia presentations is that they contain media objects requiring some notion of *time* for correct presentation. In order to model presentations, a timeline metaphor is often used. To understand SMIL 2.0, the *timeline* metaphor can be applied to containers that hold only static objects for which all timing (including activation time, termination time and object duration) are known at the time of authoring. For containers like the *excl*, where the



```
<seq>
  <img id="a" dur="6s"
    begin="0s" src="..." />
  <img id="b" dur="4s"
    begin="0s" src="..." />
  <img id="c" dur="5s"
    begin="2s" src="..." />
</seq>
```

```
<par>
  <img id="a" dur="6s"
    begin="0s" src="..."/>
  <img id="b" dur="4s"
    begin="0s" src="..."/>
  <img id="c" dur="5s"
    begin="2s" src="..."/>
</par>
```

```
<excl>
  <img id="a" dur="6s" src="..."
    begin="0s;x.activateEvent" />
  <img id="b" dur="4s" src="..."
    begin="y.activateEvent" />
  <img id="c" dur="5s" src="..."
    begin="z.activateEvent" />
</excl>
```

Figure 3: *SMIL 2.0 Temporal Groups*.

(a) shows a SMIL *seq* container, plus a SMIL 2 code fragment. Note that *begin* times are relative to the predecessor end.
(b) shows a *par* container, with the same timing offsets. Note that *begin* times are relative to the containing *par*.
(c) shows an *excl* container. Object *a* starts at time 0 of the excl *and* whenever object *x* (not shown) is activated.
Objects *b* and *c* are started only when objects *y* and *z* (not shown) are activated.
Since the actual activation times depend on event activity, a common timeline cannot be used to model the relationships among *a*, *b* and *c*.

activation or termination time is unknown until the object actually starts, or for *par*'s containing objects with unresolved *begin* or *end* times, the timeline metaphor is essentially useless. While this presents a challenge to authoring software designers, it provides a SMIL 2.0 document with unprecedented temporal flexibility and adaptability because the effective presentation timeline is determined at runtime, based on the individual media objects activated and their activation order.

Unlike media formats which force the user to explicitly define when each object begins and ends relative to a single timeline, SMIL provides a logical timing framework in which the structured relationship of objects can be used to define most timing relationships among objects. When writing a SMIL file, the author does not have to worry about the exact starting or ending times, since these can be gleaned from the presentation's *par*/*seq*/*excl* structure. In a SMIL specification, the timeline is determined by the structured composition of objects, rather than the timeline being the basic compositional unit.

The decoupling of timing resolution from presentation specification allows, for example, the entire timing of a presentation to be changed based on the dynamic associations of content. Consider the following SMIL 2.0 code fragment:

```
<par endsync=”select”>
  <img id=”btn_a” src=”...” dur=”10s” />
  <img id=”btn_b” src=”...” dur=”5s” />
  <excl id=”select”>
    <text  src=".../todays_txt.html" be-
gin=”btn_a.activeEvent” dur=”25s”/>
    <video src=".../todays_video.mpg” be-
gin=”btn_b.activeEvent” />
  </excl>
  <audio src=".../todays_tune.mp3" repeat=”indefi-
nite”/>
</par>
```

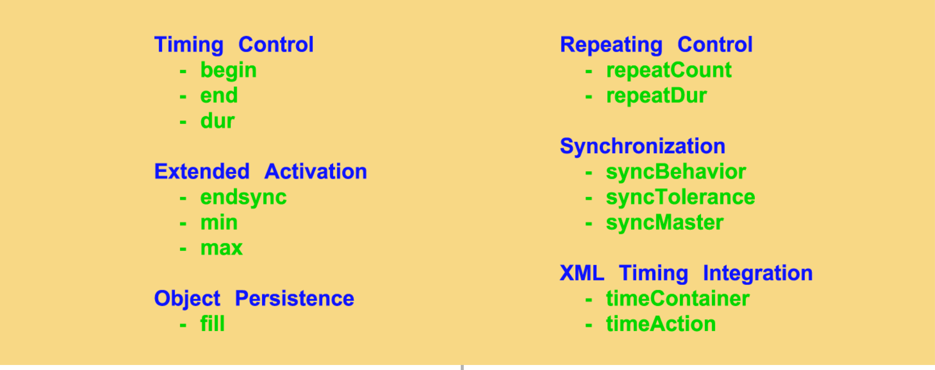
The outer *par* contains two button images (*btn_a* and *btn_b*), an audio object reference, and an *excl* container (named select). The *excl* contains a text object reference and a video object reference. Note that each of the media object references is indirect: they point to external data that can change on a hourly, daily or weekly basis.

The button images are displayed for 10 and 5 seconds, respectively, when the *par* starts. The background music also begins, and repeats indefinitely — that is, it repeats until the containing *par* ends. Since the *par* contains the directive:

```
endsync=”select”
```

it will end when the *select* object ends. The *select* object ends when either the text or the video object ends, depending on which one is activated by the presentation viewer. (If the viewer selects *btn_a*, the text object will be shown for 25 seconds; if the viewer selects *btn_b*, the video object will play for the intrinsic duration that object, whatever that is.) All the while, the background music keeps repeating.

Drawing the presentation timeline is left as an exercise to the reader.



Timing Control
- begin
- end
- dur

Extended Activation
- endsync
- min
- max

Object Persistence
- fill

Repeating Control
- repeatCount
- repeatDur

Synchronization
- syncBehavior
- syncTolerance
- syncMaster

XML Timing Integration
- timeContainer
- timeAction

Figure 4: *Primary SMIL 2.0 timing and synchronization attributes.*

SMIL 2.0 Timing and Synchronization

The timing and synchronization functional group represents the core of SMIL 2.0 functionality. The group is divided into 19 modules, each of which defines a collection of XML elements and attributes that control some aspects of timing.

The three basic SMIL 2.0 timing elements *seq*, *par* and *excl* were introduced above. Each of these elements form parent timing containers in which media objects or other timing containers can be placed. For any content within a container (whether a media object or structure container), the primary issues to be addressed are:

- when does the element begin?
- how long is it active?
- what happens to it when it is no longer active?
- other than strict temporal, are there other conditions that cause an element to end?

These questions are ‘answered’ by specifying a set of attribute values on either the parent time container or any of its children. SMIL 2.0 has an extensive set of attributes to control timing, most of which carry sane defaults so that basic timing and synchronization operations can be accomplished easily.

The basic collection of timing and synchronization attributes are shown in Figure 4. Note that not all of the SMIL 2.0 profiles support all of the attributes, and the syntax of defining and setting the attribute values may vary, but the core functionality of timing, extended activation control, object persistence and repeating element control are reasonably universal. A design objective of SMIL 2.0 was that each attribute should have a well-defined semantic that remains constant across profiles.

Begin, End and Dur

The *begin* and *end* attributes are very similar in terms of syntax and semantics. The primary differences between these attributes in SMIL 1 and SMIL 2.0 is that SMIL 2.0

allows multiple values to be specified for begin and end. (An example of this was shown in Fig. 3.) The first begin/end value that is satisfied will cause a corresponding element to start or end. It is possible to mix both scheduled and event-based activation/ termination in one attribute. For example:

```
    begin=”3s;button.activateEvent”
```

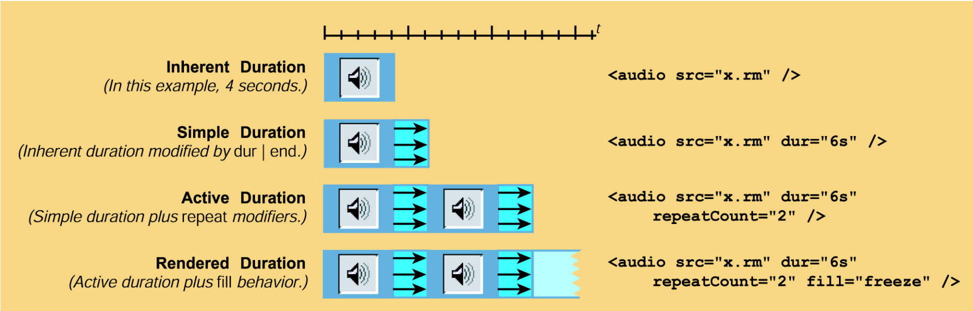
will cause the associated element to start either at 3 seconds after the default time at which the element would otherwise be able to start, or when the *activateEvent* event associated with another element with an ID of button occurs (typically via a mouse click). The ‘default’ begin time varies with the parent time container: for a *par* and *excl*, it is when the container starts; for a *seq*, it is when the previous element in the *seq* ends (or at the start of the *seq* for the first child element).

Once started, an element will have a certain duration, which may be determined in several ways (see Figure 5). In general, so-called discrete media — media without an inherent notion of time, such as images or a page of text—have a default duration of 0 seconds, while continuous media—media with an inherent notion of duration, such as an audio or video object—use that inherent duration as default. (Some implementations of SMIL 1.0 used 5 seconds for the default duration of a discrete media item. All SMIL 2.0 implementations should use the “real” default value of 0.)

An object with a duration of 0 seconds would ordinarily not be visible during a presentation. Such objects can be displayed if the *fill* attribute is set to a value of *freeze*. A frozen object is displayed after the end of its active duration until its parent time container ends — for discrete media, this will be the image or text, for continuous media, it will be the last frame/sample. The following:

```
    <par dur=”10s”>
      <img begin=”3.5s”
         fill=”freeze” />
    </par>
```

will display an image 3.5 seconds after the parent *par* begins. The active duration of the object will be 0 seconds, but the object will remain visible for 6.5 seconds (until the parent *par* ends). Two values of *fill* define a rendering duration that extends beyond the parent: *transition* and *hold*.


t
Inherent Duration
(In this example, 4 seconds.)
Simple Duration
(Inherent duration modified by dur | end.)
Active Duration
(Simple duration plus repeat modifiers.)
Rendered Duration
(Active duration plus fill behavior.)

```
<audio src="x.rm" />

<audio src="x.rm" dur="6s" />

<audio src="x.rm" dur="6s"
    repeatCount="2" />

<audio src="x.rm" dur="6s"
    repeatCount="2" fill="freeze" />
```

Figure 1: Various SMIL 2.0 duration concepts.
The *inherent duration* is the duration of the media object, if any. The *simple duration* is the *inherent duration* modified by the *dur* attribute. The *active duration* is the *simple duration* modified by the *repeatCount* (and *repeatDur*) attributes. The *perceived duration* is the visual behavior of an element after its *active duration*, before its parent time container ends.

As in SMIL 1.0, if a *par* has multiple children active, it can specify that the entire *par* ends when the first child ends, when the last child ends or when a named child ends. This is done via the *endsync* attribute. (The default is: `endsync="last"`.)

SMIL 2.0 has two new attributes that provide extra duration control: *min* and *max*. These attributes, which developed out of the SMIL 2.0/MPEG-4 integration work, can be used to define a lower or upper bound on the active duration, regardless of that element's timing characteristics.

What goes around, comes around...

The default behavior of all elements is that they play once, for either an implicit or explicit duration. The SMIL 2.0 *repeatCount* attribute allows an iteration factor to be defined that acts as a multiplier of the object's simple duration. A special value *indefinite* is used to specify that an element is to be repeated continually until the parent time container ends. Although it is tempting to define this as "forever," this is not correct: the parent time container defines the context of *indefinite*. The *repeatDur* attribute defines a duration for all of the repeated iterations.

Synchronization behavior

In a perfect world, all of the defined timing in a specification would be implemented perfectly by a SMIL 2.0 player. Unfortunately, the world is not only imperfect — it is also unpredictable. In order to provide some measure of control in the face of unpredictably, SMIL 2.0 provides three high-level synchronization control attributes: *syncBehavior* allows a presentation to define whether there can be slippage in implementing the composite timeline of the presentation; *syncTolerance* defines how much slip is allowed; *syncMaster* allows a particular element to become the "master" timebase against which all others are measured.

XML integration

When used in a native SMIL 2.0 document (one in which the outer XML tag is `<smil>`), the nature and meaning of various timing elements is clear. When integrating SMIL timing into other XML languages, a mechanism is required to identify timing containers. The SMIL 2.0 specification does this using the *timeContainer* and *timeAction* attributes.

Events and Hyperlinking

In the normal course of processing, the activation hierarchy of a SMIL 2.0 document determines the rendering of document elements. The user can influence the elements selected by using SMIL 2.0 *events*. The event architecture allows document components that are waiting to be activated or terminated to actually start or stop. There are several uses of events allowed in SMIL 2.0, but perhaps the most important new semantic in the language is the combination of events and the *begin*/*end* attributes. In further combination with the *excl* element, events provide a very powerful mechanism for conditional content activation.

SMIL 2.0 also supports a rich hyperlinking architecture. Unlike links in (X)HTML, the fundamental concept of the SMIL link is that it models a *temporal seek* in a presentation. Rather than simply activating a target element, the target play state that is activated is identical to the state that the presentation would have been in if the target point had been arrived at ‘naturally’. (One exception is that all event-based activation is ignored.) This means that all nodes temporally between the source and destination of the link need to be evaluated to see if they would have contributed to the final target play state. The temporal seeking and activation facility allows very polished presentation construction — but its implementation in the player is not for the faint-hearted!

Content Control Elements and Attributes

One of the major innovations of SMIL was support for conditional content via the *switch* element.

```
<switch>
  <video src=”...” systemBitrate=”115200” />
  <seq systemBitrate=”57344”>
      <img src=”img1.png” dur=”5s” />
      ...
      <img src=”img12.png” dur=”9s” />
  </seq>
  <text src=”desc.html” dur=”30s”/>
</switch>
```

In this fragment, a video object is rendered if the system bitrate (actual or configured) is set to 112K or above. If the bitrate is 56K or above — but below 112K — a sequence of images is shown instead. If no other element had been selected in the *switch*, the text object is shown.

One of the limitations of the SMIL 1.0 switch element was that it only allowed selection based on pre-defined list of “system” attributes. SMIL 2.0 extends this notion with a user-defined set of test attributes: *custom test attributes*. The custom test attributes can be defined by the author and selected by the user directly or indirectly via the player. SMIL 2.0 also allows test attributes to be used “in line”: that is, outside the switch element. Any media object reference containing a system or custom test attribute will be evaluated as if it were wrapped into a single-element switch.

To a first approximation, both the event mechanism and the content control facilities provide a means for dynamic selection of objects in a presentation. The basic difference between these facilities is that the event mechanism works on objects that have been recognized by the SMIL 2.0 scheduler, while the content control facility determines which object the scheduler gets to evaluate. The actual selection process associated with the conditional control primitives may be static or dynamic. That is, the selection may be done at parse time (when the document is loaded) or at each iteration through the document. The specific selection policy is a property of the SMIL player.

Transitions and Animation

SMIL 2.0 significantly extends the facilities available for performing local operations on media objects in a document. Two of the most visible of these facilities are support for *transitions* and *animation*.

SMIL’s transition support allows a set of basic transitions to be defined as part of the SMIL *head* element, and then to instance one of the available transition types as an input or output transition on a media object. For example, consider the SMIL 2.0 fragment in Figure 6. Transitions can be applied to all visual media, or collections of media. Each transition can have certain timing properties (duration) and other transition-specific properties (direction).

Animation in SMIL 2.0 comes in two flavors. First, there are animations that apply to attributes and elements in the SMIL presentation. This includes animating the position of a rendering region or a background color. A second type of animation support is the SMIL 2.0 Animation Specification, which gives generalized temporal animation support for integration into XML .

Layout

Unlike HTML, which uses an indirect layout model via CSS, SMIL also supports a direct layout model for managing the visual and audio rendering space associated with a presentation. In SMIL 2.0, the SMIL basic layout mechanism is supported with elements and attributes that allow layout to be specified as a hierarchy of rendering regions, and to be able to support multiple top-level presentation windows. The support for hierarchical layout is especially important when integrating animation into a presentation: content that is logically grouped together can be moved in concert by animating the position of the common parent region. Multiple top-level windows allow a single presentation to segment control and rendering operations in different parts of the screen in a coordinated manner.

Another enhancement to SMIL 2.0 layout is the support for sub-region positioning.This facility allows an object to be placed at a particular X,Y offset within a region or aligned to a *registration point*. (In SMIL 1.0, all objects were placed at the top left corner of a region.) The placement occurs in-line, as part of the media object reference. SMIL 2.0 also supports an alignment mechanism for content in regions; this allows a set of images of varying size to be centered at a specific point in a region. SMIL 2.0 also allows multiple objects to be active in a region simultaneously, relaxing a restriction from SMIL 1.0.

(This chapter was adapted from the book SMIL: Interactive Multimedia on the Web, *by Lloyd Rutledge and Dick Bulterman. A version of this chapter also appeared in an article in IEEE Multimedia Magazine, by Dick Bulterman, founder of Oratrix.)*

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<smil xmlns="http://www.w3.org/SMIL20/Language">
  <head>
    <layout>
    ...
    </layout>
    <transition id="fade" type="fade" dur=”1s”/>
    <transition id="push" type="pushWipe" dur=”0.5s”/>
  </head>
  <body>
    ...
    <img src=”...” transIn="fade"/>
    ...
    <video src=”...” transOut="push"/>
    ...
  </body>
</smil>
```

Figure 6: *Transitions architecture in SMIL 2.0.*
The transitions are defined in the head section and referenced in the body section as either input or output transitions.

GR*i*NS Quick Reference Information

SMIL Compliance Information

The GR*i*NS supports the entire SMIL 2.0 specification, although not all features are previewed during editing. Documents that make use of these constructs are parsed correctly, but the features are ignored during rendering.

Supported Media Table

The following chart gives a listing of the media types supported by various versions of GR*i*NS Editor for RealOne:

MIME type	**Extensions**	**Windows 98/2K/XP**
audio/basic	au	yes
audio/x-aiff	aiff, aifc, aif	yes
audio/x-wav	wav	yes
image/jpeg	jpeg, jpg	yes (2)
image/png	png	yes (1)
image/tiff	tiff, tif	yes
image/x-portable-anymap	pnm	no
image/x-portable-bitmap	pbm	no
image/x-portable-graymap	pgm	no
image/x-portable-pixmap	ppm	no
image/x-rgb	rgb	yes
image/x-xbitmap	xbm	no
	bmp	yes
	ras	yes
	tga	yes
video/mpeg	mpeg, mpg	yes
video/quicktime	qt	yes
video/x-msvideo	avi	yes
video/x-sgi-movie	mov	no
text/html (5)	html, htm	no
text/plain	txt	yes

Notes

1. Preview support seems to be somewhat buggy.
2. Not all encodings supported.
3. HTML data is rendered by the GR*i*NS Preview Player, but not converted to RealText.

Each of these formats is converted to the appropriate RealSystem datatype. For highest quality rending of a final presentation, we recommend converting some datatypes to RealMedia before inserting them in a presentation, if possible.

RealOne Media Conversion

The following chart describes the levels of support provided in the GR*i*NS/RealOne version for the listed RealMedia data types used in the RealNetworks RealOne Player:

RealMedia	**Extension**	**Importable**	**Generated**
RealAudio	ra, rm	yes	yes
RealVideo	rm	yes	yes
RealText	rt	yes	no(1)
RealPix	rp	yes	no

Notes

1. GR*i*NS/RealOne provides support for the automatic generation of simple RealText documents from immediate strings in the Editor, but it does not at present provide full RealText editing facilities. This is expected in a future release.
2. GR*i*NS/RealOne can import and convert most existing RealPix files.

References and Links

Please see the Links section of the GR*i*NS/RealOne Web site (*www.oratrix.com/*GR*i*NS).