

Quick Start Guide



GRiNS Editor for RealOne
Windows-98/2000/XP
March, 2002



© 2002 Oratrix Development bv. All rights reserved.

GRiNS Editor for RealOne *Quick Start Guide* for Windows-98/2000/XP. March, 2002.

The software described in this manual is furnished under license and may be used only in accordance with the terms of that license.

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form without the written permission of Oratrix Development bv.

GRiNS, *GRiNS Editor for Real*, *GRiNS Editor for IE*, *GRiNS Editor for 3Gwireless*, *GRiNS/Pro* and *GRiNS/SMIL* are trademarks of Oratrix Development.

RealAudio, RealVideo, RealText, RealPix, RealMedia and RealONE Player are trademarks of RealNetworks, Inc.

Windows, Windows-2000, Windows-98SE, Windows-XP and InternetExplorer are registered trademarks of Microsoft Corporation.

Important Notices

This document is the GRiNS Editor for RealOne *Quick Start Guide*. 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 GRiNS Editor for RealOne *Quick Start Guide* 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 GRiNS Editor for RealOne, version 2.2 as implemented for Windows 2000 and XP. While the look of other versions of GRiNS are slightly different because of adherence to common conventions on those other environments, the functionality described is similar for all versions of GRiNS. In order to reduce document size, only images from the Windows version have been included in this document.

We welcome your questions on GRiNS 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 GRiNS 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

Important Notices	iii
Table of Contents	v
GRiNS Editor for RealOne Distribution Roadmap	1
Distribution Package Contents	1
Look for Updated Documentation	1
GRiNS/Pro and GRiNS/RealOne	2
GRiNS/RealOne Quick Start Guide	2
Introduction to GRiNS and GRiNS/RealOne	5
Version 2.2 of GRiNS/RealOne	5
Key Features of Version 2.2 of GRiNS/RealOne	5
The GRiNS/RealOne Work Flow	7
Building Presentations with GRiNS/RealOne	9
Installing GRiNS Editor for RealOne	11
Installing GRiNS/RealOne	11
<i>Installing GRiNS/RealOne on Windows 98/2000/XP systems</i>	<i>11</i>
<i>Checking for updated GRiNS/RealOne software</i>	<i>12</i>
A Bird's-Eye View of GRiNS/RealOne	13
The Menu/Toolbar Structure	13
<i>Pull-Down Menus</i>	<i>13</i>
<i>GRiNS/RealOne Toolbars</i>	<i>15</i>
Creating Presentations Using GRiNS/RealOne	16
Important Note	16
An Overview of the GRiNS Authoring Views	17
<i>The Previewer: The Embedded Presentation Player</i>	<i>17</i>
<i>The Structured Timeline: The Logical View of a Presentation</i>	<i>17</i>
<i>The Layout View: Managing Visual and Auditory Resources</i>	<i>22</i>
<i>The Source View: Seeing the Translated Document</i>	<i>23</i>
<i>The Assets View: Managing the Objects in the Presentation</i>	<i>24</i>
<i>The Hyperlink View: Creating Order within the Chaos of Links</i>	<i>24</i>
<i>The Transitions Window: Input and Output Visual Control</i>	<i>25</i>
Using GRiNS Editor for RealOne	27

Understanding SMIL 2.0	29
Design Goals	29
SMIL: what it isn't (and what it is!)	30
Structure, Timelines and SMIL	32
<i>Timeline Definition</i>	34
SMIL 2.0 Timing and Synchronization	35
<i>Begin, End and Dur</i>	36
<i>What goes around, comes around...</i>	38
<i>Synchronization behavior</i>	38
<i>XML integration</i>	39
<i>Events and Hyperlinking</i>	39
Content Control Elements and Attributes	40
Transitions and Animation	41
Layout	42
GRiNS Edition Comparison Information	43
GRiNS Packaging Information	43
GRiNS/RealOne Quick Reference Information	49
SMIL Compliance Information	49
Supported Media Table	49
RealOne Media Conversion	51
References and Links	51
Where Next?	53

GRiNS Editor for RealOne Distribution Roadmap

Congratulations on selecting the GRiNS Editor for RealOne (GRiNS/RealOne) for creating SMIL 2.0 presentations. GRiNS/RealOne lets you harness the full power of the RealNetworks RealOne player in an easy and intuitive manner.

Distribution Package Contents

The GRiNS/RealOne distribution package consists of the following components:

- GRiNS Editor for RealOne *Quick Start Guide*: an overview of the installation instructions for GRiNS and a tour of the basics of the GRiNS Environment. (This is this guide.)
- GRiNS Editor for RealOne *Reference Manual*: a comprehensive description of the GRiNS/RealOne features and options;
- GRiNS Editor for RealOne *Template Guide*: a quick and easy description of the standard GRiNS/RealOne template set that gets you started making RealOne presentations;
- GRiNS Editor for RealOne *Tutorial/User's Guide* and *HowTo Guides*: a set of step-by-step instructions for creating a wide range of RealONE presentations.
- *Templates*: a set of templates used in GRiNS/RealOne, which you may use to build your own presentations;
- *GRiNS-Icons*: a directory containing Icons used by GRiNS/RealOne;
- *GRiNS-Demos*: a directory containing several short GRiNS demonstrators; and
- *Software*: depending on the distribution you downloaded, a GRiNS distribution for Windows-95/98/2000/XP or other platform.

Look for Updated Documentation

The documentation and demonstration presentation set for GRiNS will be expanded to describe and illustrate more GRiNS features. The most up-to-date versions of all of our manuals and demonstrators are available via the documents link at <http://www.oratrix.com/>.

GRiNS/Pro and GRiNS/RealOne

GRiNS is available in several configurations: a *Pro* edition, intended for users who want full control over authoring and publishing to many popular SMIL-2.0 engines (such as the RealNetworks RealOne player, the XHTML+SMIL implementation in Microsoft's InternetExplorer 6.0, and 3GPP/Mobile SMIL for wireless devices), and dedicated editions for particular streaming players (such as GRiNS/RealOne). Unless otherwise marked, all other features discussed in this Guide are available in both GRiNS/RealOne and GRiNS/Pro. (See section 6 for a full feature comparison.)

If you purchased GRiNS/RealOne and would like to upgrade to GRiNS/Pro, please visit the GRiNS/RealOne upgrade page at:

<http://www.oratrix.com/>

Note that your original license number will be required for a product upgrade to GRiNS/Pro.

GRiNS/RealOne Quick Start Guide

The GRiNS/RealOne *Quick Start Guide* will help you get started making SMIL presentations for the RealOne Player. It is divided into eight sections:

1. *Introduction to GRiNS and GRiNS/RealOne*: a description of the GRiNS architecture and a discussion on how you can protect your authoring investment by using the GRiNS approach to presentation design and maintenance.
2. *Installing and Running GRiNS*: an overview of how to load GRiNS/RealOne onto your system and what to expect when you load an existing RealPlayer presentation in the GRiNS/RealOne Editor.
3. *Bird's-Eye View of GRiNS*: a quick tour of the basics of the GRiNS/RealOne environment. Spending five minutes here will get you ready to create and edit complex presentations without ever having to type a single line of SMIL code.
4. *Using GRiNS/RealOne*: a simple tutorial that gets your fingers and mind working in GRiNS compatible mode.
5. *Understanding SMIL-2.0*: an overview of the basic SMIL-2.0 concepts used by all versions of GRiNS.

6. *GRiNS Edition Comparison Information*: an overview of the differences in functionality offered in the various editions of GRiNS.
7. *GRiNS Editor Quick Reference*: an overview of the data types supported by the GRiNS/RealOne environment and other useful reference information for getting started with the GRiNS/RealOne Editor.
8. *Where Next?*: a set of pointers to GRiNS/RealOne resources and other hints to help you on your way to creating presentations for the RealOne Player.

Each of the sections has been written to be relatively stand-alone, but we suggest that you read them all to get a good overview of the system.

Introduction to GRiNS and GRiNS/RealOne

GRiNS/RealOne is an authoring environment for creating, editing and maintaining streaming media documents for the RealOne player. GRiNS/RealOne allows you to take source materials such as audio, video, image and text *assets*, and integrate them into a presentation that can be distributed via the Web. GRiNS/RealOne is one product in a range of tools that help you create compelling presentations for Web-based multimedia — for more details on the GRiNS product family, see <http://www.oratrix.com/>.

Fundamental to GRiNS/RealOne is that the complex task of presentation authoring is partitioned so that you have the freedom required to build and manage presentations easily. At the “assets-end,” you can work with the media assets, management tools and editors that best meet the requirements of the task at hand. At the “presentation end,” GRiNS allows presentations to be published to Real’s optimized datatypes and uploaded to a streaming server of your choice.

The GRiNS/RealOne environment has native support for the W3C SMIL-2.0 language. This means that you can open and edit existing presentations, whether or not they were made with GRiNS. This gives you an advantage by reducing your development time: you can update earlier versions of presentations made by others, or you can create new presentations directly with GRiNS/RealOne.

Version 2.2 of GRiNS/RealOne

GRiNS/RealOne has been designed to help you produce compelling productions that are targeted directly to the RealOne player. This gives you access to the millions of users who have Real’s player installed on their systems.

Key Features of Version 2.2 of GRiNS/RealOne

GRiNS/RealOne has been based on the feedback we have received from the thousands of GRiNS users world-wide. The key features of this release are:

- integrated *Preview Player* that lets you evaluate your presentation as you create it, before you publish to RealOne;

- integrated *bandwidth simulation* for multiple levels of network performance to show if your presentation will 'fit thru the pipe';
- presentation authoring using the GRiNS/RealOne *structured timeline, layout* and *hyperlink* editing views;
- simplified link use with RealOne's *context* and *browser* windowsRealOne;
- *automatic conversion* with detail control parameters for nearly all media types into RealOne formats, including:
images:JPG, GIF, PNG, BMP formats to streaming images,
video:AVI or MPEG movies to RealVideo,
audio:AIFF, MP3 or WAV to RealAudio,
- *drag and drop* editing, using both source assets and pre-converted RealMedia files (allowing you to fine-tune asset conversion with RealProducer — or any other conversion tool — before placing it in your production);
- support for SMIL's SWITCH, allowing you to build *adaptive presentations* that can optimize your production to the available bandwidth;
- the ability to *open previously created SMIL files*, even those created with Real's Slideshow, TAG Composer, or a text editor;
- support for *interaction* via the Exclusive groups;
- support for inserting *transitions* into a presentation on all types of media;
- support for embedded *animations* on objects;
- support for *layout hierarchies*
- support for *template-based editing* (both presentation and assets templates);
- an embedded source code viewer and editor (editing in GRiNS/Pro only);
and
- easy upload to a RealSystem RealServer or other media servers (GRiNS/Pro only) for hosting your presentation.

The basis of GRiNS/RealOne is the unique GRiNS *Structured Timeline View*: you create logical presentations that can be easily edited and updated without having to worry about detailed timing interactions among or within objects.

The GRiNS/RealOne Work Flow

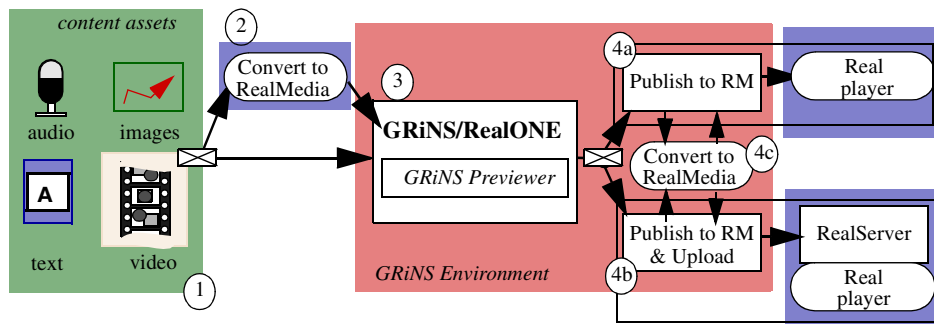
Creating compelling streaming media presentations is a complex task. The challenges that must be addressed are:

- creating or selecting appropriate media objects for the presentation
- structuring the presentation to convey the message you want
- defining a presentation that 'fits' through the network connection available to the viewer of your presentation

Designing presentations for the Web presents you with both opportunities and problems. The basic opportunity is that you can reach a world-wide audience with your message. The basic problem is that the world is a non-standard place. People speak different languages, people use different computers, people have access to the Web via connections of varying speeds.

GRiNS/RealOne helps you manage the complexity of the Web by giving you control over the essential parts of your presentation. Want to quickly get out Version 1? No problem: the GRiNS/RealOne template-based approach lets you get started in seconds. Want to make new versions that are tailored to specific groups of users, or to specific architectures or to users with high- or low-bitrate connections? Again, no problem: GRiNS/RealOne gives you the freedom to edit, update and maintain your presentations, and to target as many specific groups as you like.

The following illustration gives you an overview of the steps in the GRiNS/RealOne workflow. (Note: this illustration is geared to RealOne media flow, but is valid for other SMIL 2.0 player and server products as well.)



1. Creating content is perhaps the most challenging part of making a compelling presentation. You should keep in mind that the Web is not TV: the Web is a delivery vehicle that can vary widely in performance, while TV is a predictable delivery vehicle that 'looks good' in spite of supporting relatively low bandwidth audio and video information. In a similar vein, be careful with image data and with text: create or select each datatype carefully, keeping your intended audience, their equipment and their patience in mind. GRiNS/RealOne allows you to use nearly all popular audio, video and image formats available, including AIFF, AVI, BMP, GIF, JPEG, MPEG-1&2, MP3 and PNG.
2. An optional second step in the process is the encoding of your media assets into one of RealNetworks' RealMedia formats, such as RealAudio, RealText, RealVideo or RealPix. You can also choose to have GRiNS/RealOne do the conversion for you, but you can sometimes exercise more control over the coding of your data by using a separate tool, such as Real's RealProducerPro. GRiNS/RealOne allows you to insert both converted and unconverted objects into your presentation. (Note: playback of RealMedia is only available on RealOne systems.)
3. Once your media objects have been selected, GRiNS/RealOne's unique structure-based approach allows you to combine, edit and maintain your presentations in a flexible and easy manner. We look at this process in detail

in section 3.

GRiNS/RealOne allows you to open existing presentations for editing, whether or not initially created with GRiNS/RealOne. GRiNS/RealOne also allows you to preview your presentation incrementally, while creating it, using the GRiNS/RealOne Preview Engine.

4. Once your presentation is made, you can publish it for use with the RealOne player or, with GRiNS/Proto other SMIL 2.0 systems (such as IE-6.0 and 3GPP/Mobile). You can do this in two ways:
 - (a) publish for presentation on your local computer. Using this option allows you to playback the entire presentation using the RealOne player;
 - (b) publish and upload your presentation to a remote server. You can upload the main presentation and the associated data objects to one server and send the HTML page announcing your presentation to another location. When publishing to RealOne, GRiNS/RealOne will convert any objects to RealMedia equivalents for you (labelled 4c in our diagram), unless you already did this conversion directly in Step 2.

Using GRiNS/RealOne, you can create complex presentations from scratch, or from one of the GRiNS/RealOne templates. You can also open any existing SMIL presentation, whether created by GRiNS/RealOne, another SMIL-compatible authoring tool or a simple text editor.

Building Presentations with GRiNS/RealOne

This Guide will provide a quick tutorial that will help you get started with either the Pro or Real edition of GRiNS.

After you get your edition of GRiNS/RealOne installed, registered owners should download the expanded tutorials from the GRiNS/RealOne Web site: <http://www.oratrix.com/>.

If you would like us to showcase your work, please send a message to grins-examples@oratrix.com.

Installing GRiNS Editor for RealOne

GRiNS/RealOne is currently available on Windows-based PCs. Each implementation of GRiNS/RealOne has been developed to meet the expectations of the user community for that platform, but all versions can edit and play presentations made by any GRiNS/RealOne authoring environment.

Installing GRiNS/RealOne

The instructions for installing GRiNS depend on the platform that you use. This section reviews the general instructions. See the release notes shipped with the GRiNS software for the latest details on installing GRiNS.

NOTE:

1. The first time you execute GRiNS/RealOne, you must provide your GRiNS License Key. Instructions for using this key are provided when the key is issued.

Installing GRiNS/RealOne on Windows 98/2000/XP systems

To install GRiNS/RealOne for Windows, download and execute the file. This program will install GRiNS/RealOne and related files in the location you give during the installation process. (Note that the name of the executable may vary slightly depending on how you purchased GRiNS.)

When complete, the GRiNS/RealOne icon will be added to your desktop. The default location for starting GRiNS/RealOne is:

c:\Program Files\Oratrix\GRiNS_RV2\GRiNS_RV2.exe.

NOTES:

1. While Administrative privileges are not required to install GRiNS/RealOne, some systems may restrict your access to shared disks or certain directories. Contact your local administrator if you have problems.

2. You can remove all of the GRiNS/RealOne Components via the 'Add/Remove Programs' program in the control panel or via the Uninstall tools provided with the distribution. Administrative rights are typically required to uninstall GRiNS/RealOne.
3. The GRiNS/RealOne components have been architected to be as unintrusive to your system as possible, but care should always be taken when removing or replacing system files.
4. In the Windows release, the file extensions *.smi* and *.smil* are claimed by Real. GRiNS/RealOne places its local information in a file with the *.grins* extension.

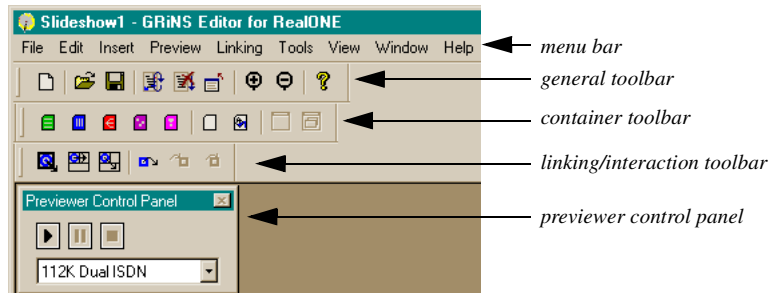
Checking for updated GRiNS/RealOne software

Once GRiNS/RealOne has been installed on your system, you may select the **Check for GRiNS update** option in the **Help** menu to see if updated versions are available.

A Bird's-Eye View of GRiNS/RealOne

The Menu/Toolbar Structure

The following menu and toolbars are available with the GRiNS/RealOne Editor:



The toolbars can be *docked* or *undocked*; you can also hide them using the View->Toolbars menu.

Pull-Down Menus

The pull-down menus contain functions that allow you to easily create and preview presentations for the RealOne player:

- The File pull-down menu allows standard file open/save/close operations. It also lets you publish your documents to RealOne and (optionally) upload them to a RealServer, or publish your presentation to XHTML+SMIL (IE-6), 3GPP/Mobile or any other SMIL 2.0 player (GRiNS/Pro only). You can also set document properties via this menu.
- The Edit pull-down menu contains standard cut/copy/paste/delete operations. It also lets you access the property tabs for a selected node. If you have a content editing program installed for the datatype of the node (such as *Adobe Photoshop* for images), you can edit the content directly with that program via the Edit Content button. (Individual content editors are not delivered or installed with GRiNS/RealOne.)

- The Insert menu lets you insert new nodes into your presentation (both structure and data nodes), and lets you wrap new structure nodes around existing ones.
- The Preview pull-down menu allows you to Start, Stop and Pause the current presentation. You can preview the entire presentation, a sub-part or a single object (either a media item or a content group).
- The Linking menu allows you to create anchors and links within your presentation, so that you can jump to them from elsewhere in the presentation or define conditional activation based on user interaction. GRiNS/RealOne provides simplified anchoring and linking support to meet common needs.
- The Tools menu lets you perform utility tasks. These include checking bandwidth usage, converting existing RealPix objects to their SMIL 2.0 equivalents, and performing specialty tasks.
- The View menu lets you manage the complexity of the display in the various GRiNS views. It also lets you visualize the bandwidth constraints of your presentation and it lets you see which objects will get played at the current simulation settings.
- The Window pull-down menu allows you to open or close the active editor views (the Preview, Structured Timeline, Layout and Source views). In GRiNS/Pro, extra facilities are offered for manipulating the presentation.
- The Help menu gives you access to the on-line help files.

In all cases, inactive selections are 'grayed-out' in the menus. Many of the sensible alternatives (based on what you are doing in the Editor) are also available in the context-sensitive commands provided under the right mouse button.




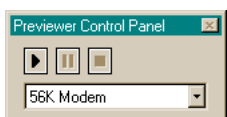
NOTES:

1. In GRiNS/RealOne, you may freely import presentations made with a text editor or with any other SMIL editor.
2. The last step in creating a presentation is typically a *Publish to Real* operation (found under the File menu); intermediate information on your presentation is saved in a file with a *.grins* type. This is a generic SMIL-compliant version of your presentation without conversions to *.rm* data types.

3. For some types of changes (like global substitutes) it is often desirable to use a text editor rather than a structure editor. This is possible with GRiNS/RealOne: simply open the *.grins* file with your favorite text editor and make whatever changes are necessary. You can then re-open the file from within GRiNS/RealOne. (Be careful when doing this, since errors put in the file by direct text editing may make the presentation unusable from within GRiNS/RealOne.) You may also inspect the source at any time with the built-in source viewer within the Editor. Note that the full-featured GRiNS/Pro version extends this functionality with a complete source editor, adding syntax and semantic checking to the editor.

GRiNS/RealOne Toolbars

GRiNS/RealOne provides four toolbars, each of which provides support for common editing tasks.

- The General toolbar provides support for opening, saving, reopening and closing *.grins* documents. It also provides controls for zooming in/out of other views.
- The Containers toolbar provides support for creating various *groups* in the presentation (par, seq, switch, excl and prior), or for creating nodes or layout regions.
- The Linking and Interaction toolbar allows short-hand creation of links to the RealONE *context* and *browser* windows and begin/end *interaction events* on groups and objects.
- The previewer control allows the preview function of GRiNS/RealOne to be activated. Any SMIL switch settings in the document can be dynamically set by the previewer, so that the document can be previewed under a range of real-world conditions.

Each toolbar icon has a tool-tip documentation string. Note that the colors of the icons in the Containers toolbar match those of the respective groups in the Structured Timeline view.

The initial state of the toolbars is that all but the previewer control is docked.

Creating Presentations Using GRiNS/RealOne

There are several key concepts that are common to all editions of GRiNS/RealOne that will help you be more productive:

- One of the GRiNS windows is the **Previewer**. This is the run-time preview of a document. While the Previewer allows a complete presentation to be viewed using conventional start/stop/pause controls, it also allows you to play sub-parts of a document — anything from a complex sub-presentation to a single media object can be activated at any time. This is a tremendous productivity enhancement tool during authoring.
- The main GRiNS/RealOne view is the **Structured Timeline**. In the Structured Timeline, you can manipulate the general structure of your presentation in a visual way. GRiNS/RealOne isolates you from the details of the SMIL language, leaving you with an intuitive means of putting together presentations based on a standard set of presentation building blocks. The Structured Timeline is the default view you get when a presentation is loaded into the Editor.
- Screen space and audio channels are managed using a concept known as the GRiNS/RealOne *Regions*.
- GRiNS provides powerful hypermedia and event-based interaction support: you can define anchors across media objects and then define various types of links between these anchors. GRiNS/Pro provides extensive features to define anchors and links at various (sub-)parts of a presentation, while GRiNS/RealOne gives you the ability to build simple links and anchors for navigation in and through your presentation.

GRiNS/RealOne also provides full support for building adaptive presentations using the SMIL switch statements. The semantics and properties of the SMIL switch are managed in the GRiNS structure view. Here again, GRiNS worries about the syntax — you simply need to define the options available.

Important Note

The SMIL 2.0 language provides many powerful primitives for creating sophisticated presentations. The more you know about SMIL 2.0, the richer your

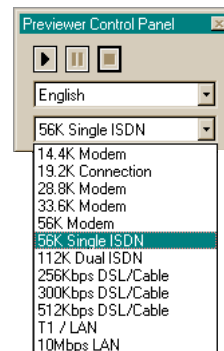
presentations will become. In order to become familiar with the basic concepts of SMIL, we suggest that you read the chapter *Understanding SMIL 2.0*, at the end of this guide. You should also read the *Real Production Guide* to understand the basic concepts of both SMIL and streaming presentations with RealOne.

An Overview of the GRiNS Authoring Views

This section will look at each of the views provided by GRiNS/RealOne and gives you an overview of how GRiNS helps you to manage the workflow during document creation and maintenance. The GRiNS/RealOne Editor was designed to let you not only build, but also edit, tailor and refine presentations — even if they weren't made with the GRiNS Editor initially!

The Previewer: The Embedded Presentation Player

One of the unique features of GRiNS is the ability to incrementally preview a presentation as it is being designed. To do this, select Preview from the Preview menu or click on the VCR-like start button on the Preview control window. If you have a switch in your presentation, the *preview control* lets you set the environment values used during previewing.

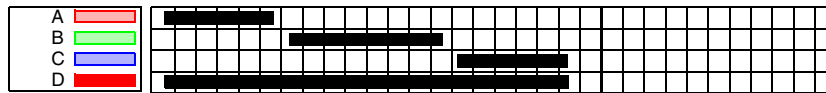


The Structured Timeline: The Logical View of a Presentation

When you open a presentation, the Editor gives you a picture of the relationship among the objects in that presentation. We call this the Structured Timeline. The Structured Timeline can be very simple for simple presentations, or it can be a complex presentation roadmap for large, complex presentations.

GRiNS takes a different approach to creating presentations than most timeline or text-based systems, so getting to know the nature of GRiNS now will help you to become more productive in the future.

The following illustration is typical of most timeline-based editors:



In this approach, a set of media items is listed on the left and their activation times are shown on the time graph. Pretty simple. Unfortunately, the time graph doesn't give us essential information. First, the length of the bottom line (node D): is it 20 time units long because this is the sum of the times of A, B and C — or are A, B and C 20 seconds in total because of D? What happens if we substitute B with an item that is 12 units long — should C shorten its duration (because of D), should B and C overlap, and should the duration of D be adjusted? Also: what if we want to show a video on a high-bandwidth connection and a sequence of four images on a low-bandwidth connection — how do you draw *that*?

Timeline systems actually hide the logical relationships among items. This means that editing, updating and maintaining a presentation is a pain in the fingers of the designer. You can't reuse combinations of media, and rather than having the timeline automatically created for you, you have to do it all by hand.

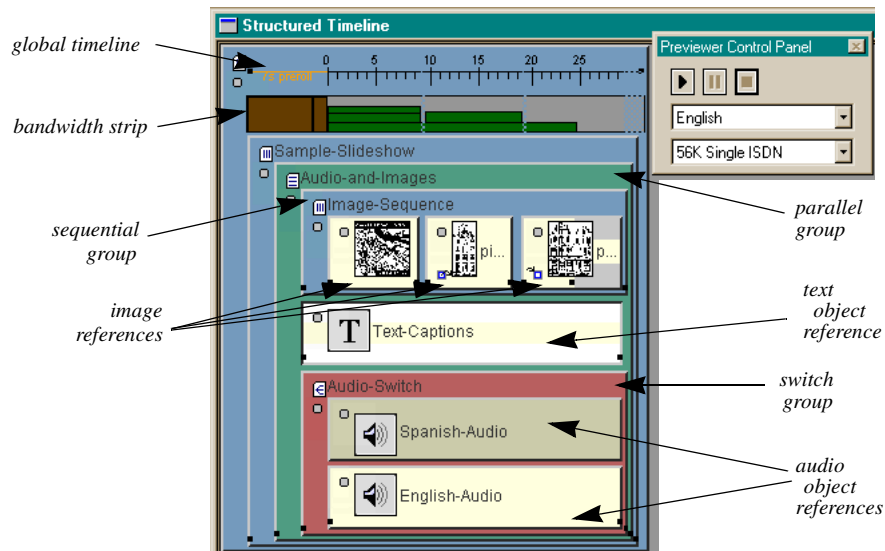
The editing philosophy behind GRiNS is that of *structured authoring*. GRiNS creates documents based on the structural relationships among components rather than getting lost in the details of a single object's activation. This not only helps initial creation, it also makes editing and updating a document much easier.

For example, if a single presentation contains an audio node and an image node that are to be presented together, GRiNS puts these inside a *parallel structure group*. If, on the other hand, a presentation contains two audio nodes that get played one after the other, GRiNS puts these inside a *sequential structure group*. If you want to define a set of alternatives (such as having both Spanish and English audio tracks available in a presentation), GRiNS puts these inside of a *switch group*. All of these notions reflect the way that SMIL works: the parallel group is equivalent to SMIL's PAR, the sequential group to SMIL's SEQ and the switch group to — you guessed it! — SMIL's Switch. (This is no accident: the designers of GRiNS/RealOne were also the co-designers of SMIL.)

Consider the following SMIL code fragment:

```
<par id="Audio and Images">
  <seq id="ImageSequence">
    
    
    
  </seq>
  <text id="Text-Captions" src="d.txt" fill="freeze"/>
  <switch id="AudioSwitch">
    <audio id="SpanishAudio" src="esp.wav" systemLanguage="Spanish"/>
    <audio id="EnglishAudio" src="eng.wav"/>
  </switch>
</par>
```

The following annotated view shows the basic GRiNS version of the same file.



This picture describes a presentation named *Sample Slideshow*. It contains three images played in sequence, which are presented in parallel with one Text object containing text comments and either an English-language or Spanish-language audio object (the selection of which will be determined by the Player).

Please re-read the last paragraph and look closely at the image: if you understand the correspondence, then you already understand the basics of GRiNS.

Unlike a standard timeline, the GRiNS Structured Timeline shows the structure containers that define the logical relationships around the media. With a bit of practice, this gives you the full logical and temporal story in one view.

The purpose of the Structured Timeline is to give a logical projection of the contents in a presentation to the author. The Structured Timeline contains the following types of information:

1. It shows the *media objects* used in the presentation; these are pointers to a particular object located either in a local file or as a remote asset on the Web or in a database, and its associated properties for this particular use in the presentation.
2. It shows what kind of *structure groups* are used to compose the presentation. These structure groups can contain a nested combination of data objects or other structure groups.
3. It shows a general presentation timeline. For this simple presentation, in which all times are resolved (that is, there is no interaction), the timeline matches the passage of time as seen by the viewer. (This is not always the case.)
4. It shows a *bandwidth usage strip*. The strip shows when objects are loaded, and if enough bandwidth exists to execute the presentation. It also shows preload time and stalls. Note that **bandwidth is not an abstract concept**. The *player control* allows you to set a particular presentation bandwidth and see how your presentation will behave for that class of connection.

Each of the basic types of container objects used in the Structured Timeline are color coded for easy recognition. With a bit of practice, you can build a mental picture of your presentation without ever hitting Preview — but since GRiNS lets you play anything from a single node to an entire presentation during editing, save those brain cells for other activities!

You can perform nearly all of the editing operations required to create most basic presentations from within the Structured Timeline. You can add, move, cut, copy, and paste objects (both data objects and structure containers) using the menu options or right-mouse short-cuts. You can define SMIL switch groups and identify the alternatives associated with the switch. You can create basic hyperlinks between nodes in the presentation. You can zoom in/out of parts of the Structured

Timeline. You can also access all of the properties of each of the objects in the Structured Timeline for detailed control of the presentation.

When possible, a thumbnail image of either the contents or the type of data objects is shown. If the view becomes too crowded, these thumbnails are scaled or removed. (You may influence this behavior in GRiNS/Pro.)

You can select any object in the Structured Timeline and then select the Preview single object function from the Preview menu to preview that object (or structure container). If you select Play from object instead, the presentation will start at that point and continue playing through the rest of the presentation's objects. To view the entire presentation, select the Preview option from the Preview menu.

The Structured Timeline is the default Editor view. It can be activated by selecting Structure view from the Window menu. The Structured Timeline can be closed by using standard windows techniques for your platform.

NOTES:

1. There are often several ways of selecting GRiNS editing functions: you can use the pull-down menus across the top of the Editor, you often can use the context menus under the right mouse button or you can make use of the keyboard equivalents for most commands.
2. The keyboard equivalents have been customized to meet the expectations of experienced users on each of the platforms upon which GRiNS is implemented. The structure of the menus is constant across all versions of GRiNS in as far as they are relevant for those platforms.

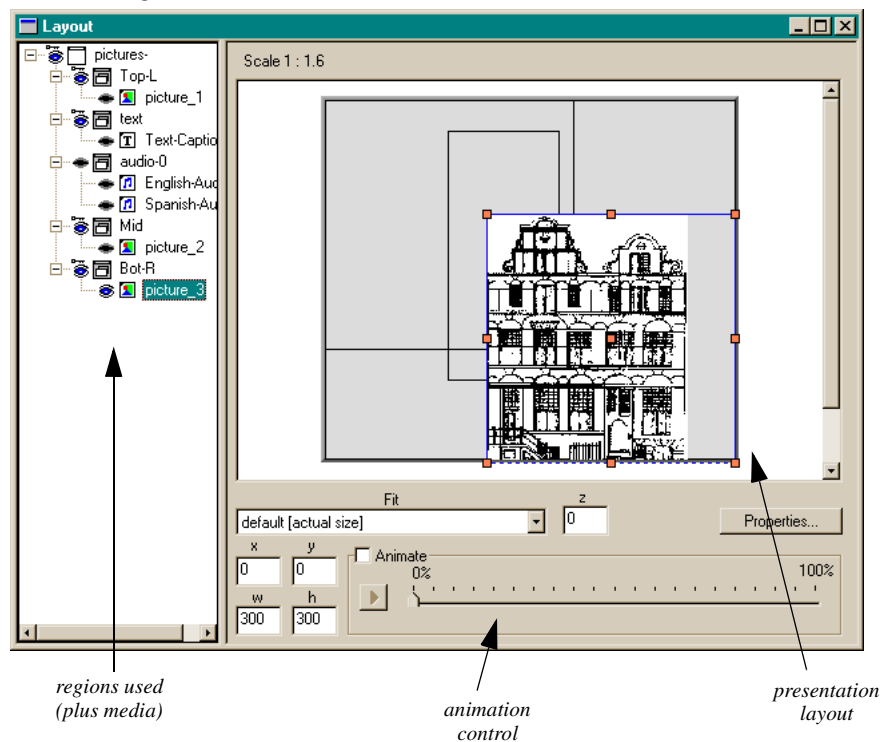
GRiNS Use Tip:

If you select Preview->Preview Single Node for an image or text file, the image/text will only appear if its duration is explicitly defined in the node.

The Layout View: Managing Visual and Auditory Resources

When working with template-based presentations, the template contains a number of Regions to organize screen and other resources in the presentation. If you use these regions, you will typically not need to visit the Layout view. For more complex presentations, or when creating your own templates, the Layout view provides a means of creating, sizing, editing and placing new regions.

The following illustration shows an annotated Layout view.



When activated, the Layout view shows a window similar to the GRiNS Player window and a separate editing window for selecting layout regions. You may select media objects for layout previewing, and you may animate objects.

The Source View: Seeing the Translated Document

In GRiNS/RealOne, you can view the SMIL source for your presentation at any time by selecting **Window->Source**. You can search for particular strings or look at the translated view of the document.

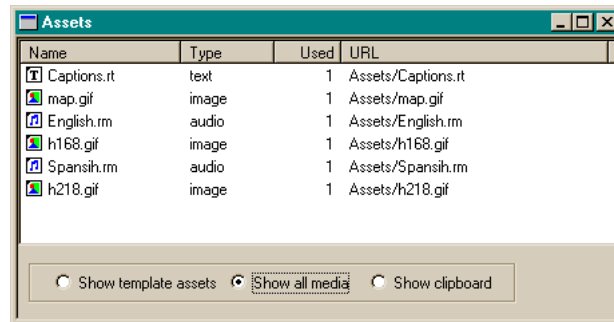


```
<?xml version="1.0" encoding="ISO-8859-1"?>
<smil xmlns="http://www.w3.org/2000/SMIL20/CR/Language" xmlns:rn="http://features.real.com/2001/SMIL20/Extensions">
  <head>
    <meta name="title" content="pictures"/>
    <meta name="base" content="file://C:/ufs/dcab/GRiNS%20Release/MSG-2.0R"/>
    <meta name="generator" content="GRiNS Pro for SMIL 2.0, v2.0 alpha win32 build 93"/>
  </head>
  <body>
    <seq id="Sample-Slideshow">
      <par id="Audio-and-Images">
        <seq id="Image-Sequence">
          
          
          
        </seq>
        <text id="Text-Captions" region="text" type="text/plain" dur="29.5s" src="Assets/Captions.rtf"/>
        <switch id="Audio-Switch">
          <audio id="English-Audio" region="audio-0" src="Assets/English.rm" systemLanguage="en"/>
          <audio id="Spanish-Audio" region="audio-0" src="Assets/Spansih.rm" systemLanguage="es-mx"/>
        </switch>
      </par>
    </seq>
  </body>
</smil>
```

(In the GRiNS/Pro version of the editor, the source view is also fully editable. Any changes you make to the document will be syntax-checked and applied against the SMIL 2.0 DTD for correctness. Integrated error feedback is also given in GRiNS/Pro.)

The Assets View: Managing the Objects in the Presentation

The Assets Window allows you to track use of individual media objects. It also allows you to manage the presentation clipboard, and it allows a template author to create structure templates that contain all types of predefined assets for use in a presentation. (We look at this in detail in the *Tutorial Guide*.)



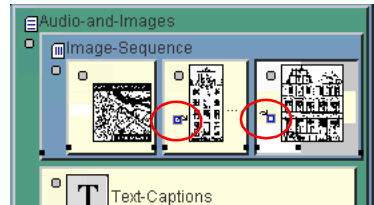
The Hyperlink View: Creating Order within the Chaos of Links

The GRiNS/RealOne Editor provides you with full control over the RealONE *Browser* and *Context* windows. Using simple short-cut icons from the Linking toolbar, you can take any media item and specify that an HTML page gets synchronized in other RealONE windows. (We explain how to do this in the next chapter.)

You can also use GRiNS to make links to other SMIL documents via *external links*. The GRiNS/Pro upgrade provides further facilities to create internal links. In GRiNS/RealOne, you can create anchors that cover an entire media object for its full activation time: these are *whole node* anchors. In GRiNS/Pro, you can create whole node anchors, plus *partial node* anchors that are active on only a part of a object (that is, over a hot spot somewhere over an image or video). You can also create *timed anchors*, that are active for only a particular part of the activation time of an object.

GRiNS/RealOne also provides controls to make simple interactive presentations using SMIL 2.0 events. Two symbols identify the presence of events:

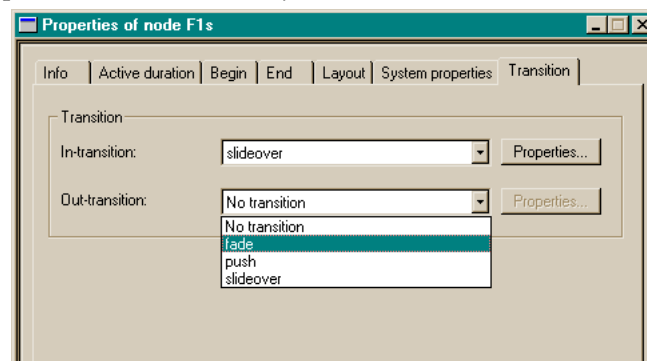
- the *event source* icon with the arrow pointing out from the center of an event box, which indicates that this node contains an event that is the source of interactive behavior in another node.
- the *event destination* icon with the arrow pointing into the center, which indicates that this node contains an anchor that is the source of a link.



You can select either icon to see the *event arc* relationship.

The Transitions Window: Input and Output Visual Control

Each object in a presentation may have one or more visual transitions associated with it. These transitions are defined in the presentation template, and can be selected by going to the properties tab for the object on which you wish to define the transition. If you select the transitions tab, you can select and set either the input or output transition for the object.



For each of the selected transitions, you can select further properties of that type (such as direction or duration).

Using GRiNS Editor for RealOne

The GRiNS distribution contains a short tutorial to get you started with GRiNS/RealOne. Please read this tutorial (one of our *HowTo* series of documents) before you start using the editor. The name of the tutorial is: *HowTo-Basic.pdf*.

January 2002

How to:

Make Template-Based Slideshows

using GRiNS/RealOne

GRiNS/RealOne “How To” Notes

(c) 2002, Oratrix, Amsterdam, The Netherlands

Overview

This document will give you all the information you need to create slideshows for the RealOne player using the GRiNS Editor for RealOne software (hereafter called simply: GRiNS/RealOne). You will learn how to create simple slideshows and how to integrate transitions, animations and links to the RealOne media browser and related infopanels. You will also learn how to publish your presentation for use with the RealOne player and a RealSystem streaming server.

This is one in a series of GRiNS/RealOne *How To* guides. While it is not intended to be a substitute for reading the GRiNS *Quick Start Guide*, the *Template User's Guide* or the *Reference Manual*, it does provide enough information to get you up and running with GRiNS quickly and easily.

Your comments, suggestions and criticisms of this document and the GRiNS environment are welcome. Please send all comments to:
grins-support@oratrix.com

In the case of error reports, please include a brief description of the process that led to the error, so that we can repeat it.

Table of Contents

<i>The Basics</i>	2
Obtaining the GRiNS/Pro Editor	2
GRiNS/Pro Workflow	2
<i>Creating Basic Slideshows</i>	3
Open GRiNS/Pro and Select a Slideshow Template	3
Working with the Structured Timeline	3
Add Media Assets to the Presentation	4
Making Adaptive Presentations	6
Preview the Presentation using the GRiNS Previewer	8
<i>Polishing the Presentation</i>	8
Adjusting Presentation Timing	8
Adding Transitions to Media Objects	10
Adding Animation	10
Selective Object Positioning	12
Synchronizing Content with the RealOne Media Browser Pane	12
<i>Generate/Upload the presentation</i>	13
Check Bandwidth Usage	13
Converting Media Objects	13
Creating the SMIL File	13
Upload to a Media Server	14
<i>Advanced Topics</i>	15

Understanding 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

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.

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 GRiNS/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 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.

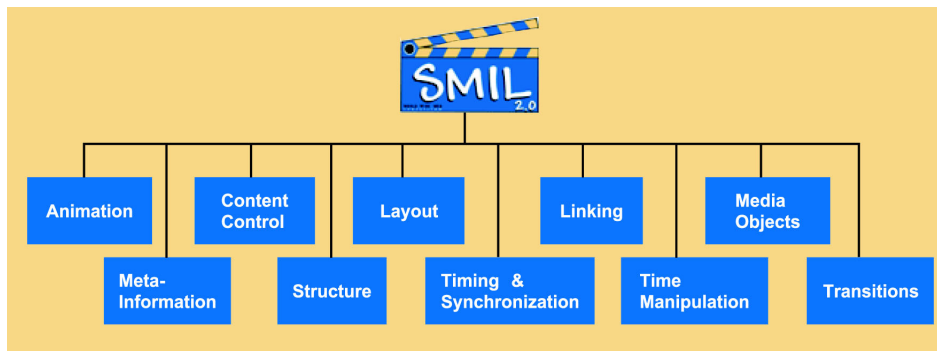


Figure 1: SMIL 2.0 Functional grouping of module sets.

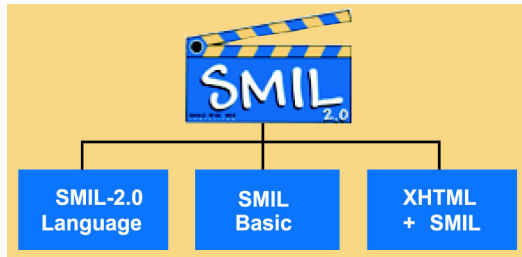


Figure 2: *SMIL 2.0 Profiles*.

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

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.

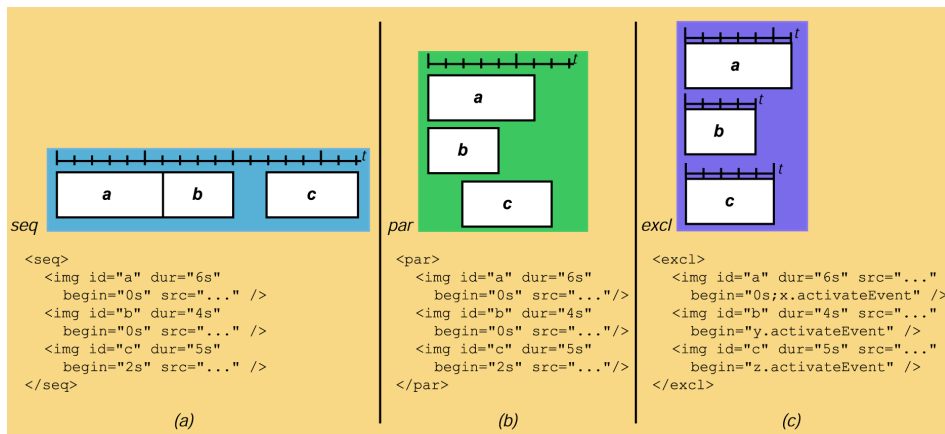


Figure 3: SMIL 2.0 Temporal Groups.

(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*.

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 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">
  
  
  <excl id="select">
    <text src="../../../todays_txt.html" begin="btn_a.activeEvent" dur="25s"/>
    <video src="../../../todays_video.mpg" begin="btn_b.activeEvent" />
  </excl>
  <audio src="../../../todays_tune.mp3" repeat="indefinite"/>
</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.

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?

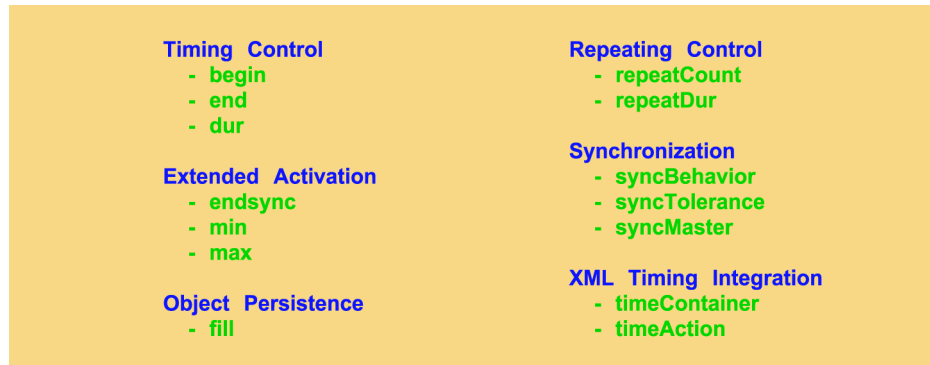


Figure 4: Primary SMIL 2.0 timing and synchronization attributes.

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

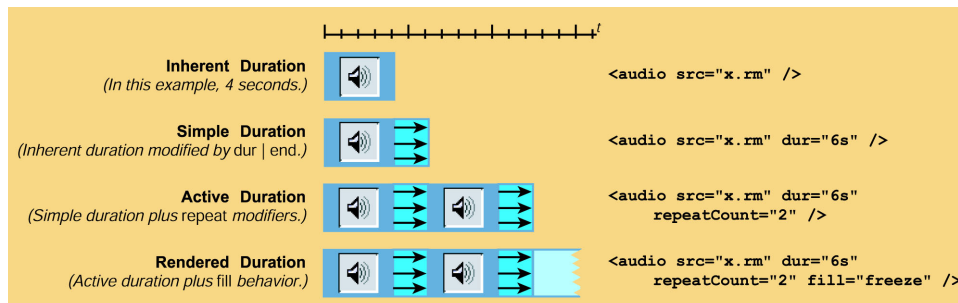


Figure 5: Various SMIL 2.0 duration concepts.

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.

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">
  
</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*.

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">
    
    ...
    
  </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*.

```
<?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>
    ...
    
    ...
    <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.

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 following 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 languages.

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.)

GRiNS Edition Comparison Information

GRiNS Packaging Information

The GRiNS Editor for RealOne is one member of a family of editing products for the SMIL 2.0 standard. This section provides a feature differentiation and compatibility matrix for the GRiNS/RealOne and GRiNS/Pro editions.

The features of each product are subject to change without notice.

GRiNS Feature Table	GRiNS / Pro	GRiNS / RealONE
Standard Features		
Structured Timeline Editing	☑	☑
Layout Editor	☑	☑
Animation Editor	☑	☑
Transition Selector/Editor	☑	☑
Source View/Editor (see below)	☑	☑
Full GUI-based editing	☑	☑
Design from Templates	☑	☑
Create custom presentations	☑	☑
Import existing presentation	☑	☑
Publish to popular players	☑	☑
Create par/seq/excl/switch groups	☑	☑
Integrate audio/video/text/images	☑	☑
Drag&Drop editing	☑	☑
Infinite Undo Stack	☑	☑
Create (timed) links	☑	☑
Insert events	☑	☑
Preview Presentation	☑	☑
Preview Individual Node/Group	☑	☑
Preview from any Node/Group	☑	☑
Model Network performance	☑	☑
Preview for different bitrates	☑	☑

GRiNS Feature Table	GRiNS / Pro	GRiNS / RealONE
New Presentations:		
From Templates	☑	☑
Custom Document	☑	☑
Create New Template	☑	--
Import:		
From Existing SMIL-2	☑	☑
From Existing SMIL-1	☑	--
From Existing RP-6/7/8	☑	☑
Publish:		
To SMIL-2 (Generic SMIL-2)	☑	--
To RealOne	☑	☑
To SMIL-1	☑	--
To RP-6/7/8	☑	--
To HTML+Time (IE-6)	☑	--
To 3GPP/Mobile	☑	--
Publish & Prune to Profile		
Based on Bitrate	☑	--
Based on Language	☑	--
Based on Captions	☑	--
Based on Any Test Attribute	☑	--
Convert		
RealPix to SMIL 2.0	☑	☑
SMIL 2.0 to RealPix	☑	--
Media Objects to RM	☑	☑
Media Objects to WMP	☑	--
Content		
Link to Installed 3rd Party Editors	☑	☑
Embedded image trimmer	☑	--
TWAIN object import	☑	--

GRiNS Feature Table	GRiNS / Pro	GRiNS / RealONE
Toolbars		
Standard Configuration	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Customize Configuration	<input checked="" type="checkbox"/>	--
UI Features		
Show Asset Thumbnails	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Customize Asset Thumbnails	<input checked="" type="checkbox"/>	--
Show Playable in Switch	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Customize 'Show Playable'	<input checked="" type="checkbox"/>	--
Structured Timeline View		
Standard Timeline View	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Customized Timeline View	<input checked="" type="checkbox"/>	--
Optimized Structured Timeline	<input checked="" type="checkbox"/>	--
Source View		
View Source	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Edit Source	<input checked="" type="checkbox"/>	--
Animation Editor		
Child Animations on Nodes	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Animations on Regions	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Full Animation Elements	<input checked="" type="checkbox"/>	--
Exclusive Nodes		
Create EXCL	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Create PRIORITY CLASS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Transitions		
Use Standard GRiNS set	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Edit Standard GRiNS set	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Define new transitions	<input checked="" type="checkbox"/>	--

GRiNS Feature Table	GRiNS / Pro	GRiNS / RealONE
Switch Support		
Specify Bitrate	☑	☑
Specify Language	☑	☑
Specify Captions	☑	--
Specify Audio Desc	☑	--
Specify Subtitle/Overdub	☑	--
Specify Screen/Depth	☑	--
Specify CPU / OS	☑	--
Specify Player/Req/Comp	☑	--
Custom Tests		
Play Custom Tests	☑	--
Define Custom Tests	☑	--
Tools		
Summary Bandwidth Usage	☑	☑
Detailed Bandwidth Strip	☑	☑
Align/Distribute Nodes in Layout	☑	--
Layout Support		
Define root-layout via UI	☑	☑
Edit root-layout numerically	☑	--
Define multiple topLayout via UI	☑	--
Edit topLayout numerically	☑	--
Define regions via UI	☑	☑
Edit regions numerically	☑	--
Define reg hierarchy via UI	☑	☑
Edit reg hier. numerically	☑	--
Define sub-regions via UI	☑	☑
Edit sub-region numerically	☑	--
Select regPoints via UI	☑	☑
Define regPoint numerically	☑	--

GRiNS Feature Table	GRiNS / Pro	GRiNS / RealONE
Timing Specification		
Specify Duration	☑	☑
Specify EndSync	☑	☑
Specify Fill	☑	☑
Specify repeatCount	☑	☑
Specify repeatDuration	☑	☑
Specify Min	☑	--
Specify Max	☑	--
Begin/End Specification		
Delay / NodeID / Indefinite	☑	☑
AccessKey / WallClock	☑	--
Event	☑	☑
X-Path Based Events	☑	☑
Restart Semantics	☑	--
Media		
via Drag&Drop	☑	☑
via local name	☑	☑
via URL	☑	☑
ClipBegin / ClipEnd	☑	☑
Event Sensitivity	☑	☑
Erase	☑	--
Synchronization Behavior		
Specify Sync Default	☑	--
Specify Sync Behavior	☑	--
Specify Sync Master	--	--
Productivity Enhancements		
Specify Default Layouts for Groups	☑	☑
Specify Default Timing for Groups	☑	--
Specify Default Fill for Groups	☑	--
Specify Default Transitions	☑	--
Merge Nodes with Parents	☑	--

GRiNS Feature Table	GRiNS / Pro	GRiNS / RealONE
Linking		
Create whole-node anchors	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Create partial-node anchors (X/Y)	<input checked="" type="checkbox"/>	--
Create timed anchors	<input checked="" type="checkbox"/>	--
Create external anchors	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Create internal anchors	<input checked="" type="checkbox"/>	--
Create Link to RealONE Browser Window	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Create Link to RealONE Context Window	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Events		
Create short-cut events	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Create/edit begin events	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Create/edit end events	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

NOTE:

1. The available versions of the GRiNS/RealOne products are updated regularly. Please consult the following Web URL for the most recent version of the GRiNS product matrix at:
<http://www.oratrix.com/>

GRiNS/RealOne Quick Reference Information

SMIL Compliance Information

GRiNS/RealOne 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 the Win32 versions of GRiNS Editor for RealOne:

MIME type	Extensions	Windows 98SE/2K/XP
audio/basic	au	yes
audio/x-aiff	aiff, aifc, aif	yes
audio/x-wav	wav	yes
image/jpeg	jpeg, jpg	yes
image/png	png	yes (2)
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

MIME type	Extensions	Windows 98SE/2K/XP
	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. Uncompressed WAV only.
2. Support seems to be somewhat buggy.
3. Not all encodings supported.

Each of these formats is converted to the appropriate RealSystem datatype. For highest quality rendering 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 GRiNS/RealOne version for the listed RealMedia data types used in the RealNetworks RealOne Player:

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

Notes

1. GRiNS/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. GRiNS/RealOne can import and convert most existing RealPix files. Limited support is available for creating RealPix from SMIL 2.0 constructs for users wishing backward compatibility.

References and Links

Please see the Links section of the GRiNS/RealOne Web site (www.oratrix.com/GRiNS).

Where Next?

The GRiNS Editor for RealOne *Quick Start Guide* provides you with a general overview of GRiNS and a particular introduction to GRiNS/RealOne.

Once you have installed GRiNS you can either set out on your own in the process of creating multimedia presentations, or you can save your time and energy by using a set of simple and straight-forward GRiNS/RealOne templates. Instructions for using these can be found in the *GRiNS/RealOne Template Guide*, which is available in on-line (HTML) or off-line (PDF) formats. Consult the documentation provided with your GRiNS/RealOne Editor access key to find the current set of templates for your version of GRiNS.

You should also consult the RealOne IQ Production Guide, available from RealNetworks, Inc. This guide provides an overview of the facilities supported by the RealOne player. It also provides a comprehensive introduction to the concepts of streaming media and details on the SMIL 2.0 language.

You should also consult profile documentation from Microsoft and the 3GPP consortium for working with other versions of SMIL 2.0. Search the Microsoft knowledge base with the keyword time2 for XHTML+SMIL, and use a Web search engine with keywords "3GPP PSS4" for information on 3GPP/Mobile.

The GRiNS Web site (www.oratrix.com/) provides a set of *release notes* and *issues reports* for each version of the GRiNS Editor and Player releases. This site also contains a publicly-available version of the GRiNS Frequently-Asked Questions list and errata to this and other GRiNS publications. If you purchased GRiNS, you will have had the opportunity to enroll for our automatic notification service for the version(s) of GRiNS you acquired.

We recommend that you use the Check for GRiNS update button in the Help menu. This version will poll the GRiNS Web site to see if an updated version of GRiNS exists. We welcome your comments, criticisms, compliments and suggestions. You can reach us at: grins-documentation@oratrix.com.

We hope you find working with GRiNS a productive experience!

