



## ***GRiNS/SMIL Tutorials Guide***

GRiNS Editor for SMIL, Version 1.5

Windows, Macintosh and Linux



2000 Oratrix Development bv. All rights reserved.

GRiNS Editor for SMIL, Version 1.5 Tutorials Guide for Windows, Macintosh and Linux. June, 2000.

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 *RealSystem G2*, *GRiNS/G2*, *GRiNS/G2Pro*, *GRiNS/G2Lite* and *GRiNS/SMIL* are trademarks of Oratrix Development.

RealAudio, RealVideo, RealText, RealPix, RealMedia and RealSystem G2 are trademarks of RealNetworks, Inc. Windows, Windows-NT, Windows-95, Windows-98 and Netshow are registered trademarks of Microsoft Corporation. Mac, Macintosh, MacOS and Quicktime are trademarks of Apple Computer Corporation. IRIX and O2 are trademarks of Silicon Graphics, Inc. Solaris and SunOS are trademarks of Sun Microsystems, Inc. UNIX is a registered trademark licensed through X/Open Company.

## Important Notices

This is the GRiNS Editor for SMIL, Version 1.5 *Tutorials Guide*. All of the information has been verified, but incremental product updates may impact part of this guide.

---

The GRiNS Editor for SMIL, Version 1.5 *Tutorials 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 SMIL, Version 1.5 distribution for Windows-95|98|NT-4 (1.5-win32.0). 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 comments on GRiNS Editor for SMIL, Version 1.5 and suggestions on this documentation. Please submit all questions and comments to our support desk at [grins-support@oratrix.com](mailto:grins-support@oratrix.com). We maintain a list server dedicated to sharing experiences among GRiNS/SMIL 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](mailto:grins-examples@oratrix.com).



## Table of Contents

<b>Important Notices .....</b>	<b>iii</b>
<b>GRiNS Tutorials Roadmap .....</b>	<b>ix</b>
Distribution Package Contents .....	ix
GRiNS Tutorials Guide .....	x
<b>Introduction .....</b>	<b>1</b>
Things to Keep in Mind .....	2
The GRiNS Workflow .....	3
A Tutorial Introduction to .....	4
A Note on the Illustrations and Conventions Used in this Guide .....	4
<i>Microsoft Windows</i> .....	5
<i>Apple Macintosh</i> .....	5
<i>Linux, SGI and SUN UNIX notes</i> .....	5
<b>Tutorial 1: Basic GRiNS/SMIL Editing &amp; Uploading Skills .....</b>	<b>7</b>
Overview and Goals .....	7
Opening a Template File .....	8
Inserting the Media Objects .....	9
Previewing the Presentation .....	10
<i>Playing the entire presentation</i> .....	10
<i>Playing part of a presentation</i> .....	10
Publishing the Presentation .....	11
<i>Check Bandwidth Usage</i> .....	11
<i>Generate the Presentation</i> .....	12
<i>Generate the Presentation and Upload</i> .....	12
Additional GRiNS Features .....	13
<i>Understanding the Structure View</i> .....	13
<i>Changing attributes of GRiNS Objects</i> .....	13
Closing Comments on Tutorial 1 .....	15
<b>Tutorial 2: Extending an Existing Presentation .....</b>	<b>17</b>
Overview and Goals .....	17
Creating the Presentation .....	18

<i>Open the Walk.grins presentation</i> .....	18
<i>Revisiting the GRiNS Structure View</i> .....	19
<i>Adding a new media object reference</i> .....	20
<i>Manipulating Structure and Data Objects</i> .....	22
<i>Previewing the Results</i> .....	23
Fine-Tuning the Timing of a Presentation .....	24
<i>Setting an Object Reference to “Loop”</i> .....	24
<i>Setting an Indefinite looping period</i> .....	25
Advanced GRiNS Features .....	27
<i>Introduction to the Timeline View</i> .....	27
<i>Fine-Tuning via Synchronization Arcs (Sync Arcs)</i> .....	28
<i>More on Timing in Parallel and Sequential Structure Nodes</i> .....	31
<i>Selecting Layout Regions</i> .....	32
Closing Comments on Tutorial 2 .....	34

### **Tutorial 3: Creating Templates, Working With Regions and Using GRiNS Resource Channels .....**

<b>GRiNS Resource Channels</b> .....	<b>35</b>
Overview and Goals .....	35
An Introduction to Layout Manipulation in GRiNS and SMIL .....	35
<i>Layout and the Walk.grins presentation</i> .....	36
Creating the Presentation .....	38
<i>Define Regions for Displaying Presentation Components</i> .....	39
<i>Define SMIL Regions via GRiNS Resource Channels</i> .....	40
<i>Defining the Structure and Contents of the Presentation</i> .....	45
<i>Previewing an Empty Presentation</i> .....	50
Closing Comments for Tutorial 3 .....	51

### **Tutorial 4: Working With Transitions and RealMedia ..... 53**

### **Tutorial 5: Adapting Contents to Meet the Needs of the User or System**

<b>Environment</b> .....	<b>55</b>
Overview and Goals .....	55
An Introduction to Adaptive Presentations in GRiNS and SMIL .....	56
Creating the Presentation .....	58
<i>Selectively Enabling a Media Object</i> .....	58
<i>Viewing Selective Activation in the Timeline View</i> .....	60

<i>Specifying Alternatives for a Group of Media Objects via the Switch</i> .....	61
Implications for Timing, Synchronization and Resource Use .....	64
Closing Comments for Tutorial 5 .....	64
<b>Tutorial 6: Working With Anchors, Links and Hypermedia</b> .....	<b>65</b>
Overview and Goals .....	65
An Overview of Linking Concepts in Hypermedia .....	65
<i>HTML Anchors and Links</i> .....	66
<i>SMIL and GRiNS/SMIL Anchors and Links</i> .....	66
<i>Using Anchors and Links in Presentations</i> .....	67
Opening the Presentation .....	68
<i>Planning the Links and Anchors</i> .....	69
Creating Simple Links in a Presentation .....	69
<i>Limitations of Simple Links</i> .....	70
Extended Linking Facilities .....	71
Closing Comments for Tutorial 6 .....	73
<b>GRiNS/SMIL Quick Reference Information</b> .....	<b>75</b>
SMIL Compliance Information .....	75
Supported Media Table .....	75
References and Links .....	77
<b>GRiNS Edition Comparison Information</b> .....	<b>79</b>
Packaging Information .....	79





## GRiNS Tutorials Roadmap

Thank you for downloading the GRiNS Editor for SMIL, Version 1.5 (GRiNS/SMIL) toolset for creating SMIL presentations. This publication will help you understand how GRiNS/SMIL works, and how it can help you to create high-quality multimedia presentations for the Web easily and effectively.

### Distribution Package Contents

Before you can create the presentations referenced in these tutorials, you must download and install a copy of the GRiNS/SMIL environment for the platform you use. You can obtain GRiNS via our electronic software resellers or via the GRiNS Web site: <http://www.oratrix.com/GRiNS>. A license is required to use the GRiNS Editor for SMIL, Version 1.5.

The GRiNS software distribution package consists of the following components:

- *GRiNS/SMIL Quick Start Guide*: an overview of the installation instructions for GRiNS and a quick tour of the basics of the GRiNS Environment;
- *Data*: a collection of data assets used in the *Quick Start* and the *Tutorial Guide* examples. You may use these when constructing your own presentations, or you may substitute your own favorite objects;
- *Templates*: a set of templates used in the *GRiNS Tutorial Guide* and which you may wish to use to build your own presentations;
- *Examples*: a collection of SMIL demonstrations;
- *GRiNS-Icons*: a directory containing Icons used by the GRiNS Editor;
- *Software*: depending on the distribution you purchased, a GRiNS distribution for Windows-95/98/NT, the Apple Macintosh or Linux;
- *GRiNS/SMIL Tutorials Guide*: a step-by-step introduction to using GRiNS. (This is the guide you are now reading.);
- *Tutorials*: a directory containing example SMIL files used in the tutorials;
- *Assets*: a sub-directory of *Tutorials* containing media objects.

We recommend that you read the *GRiNS/SMIL Quick Start Guide* before going through the tutorials in this guide.

## GRiNS Tutorials Guide

The GRiNS Editor for SMIL, Version 1.5 *Tutorials Guide* will help you learn how to make SMIL presentations easily and quickly. It is divided into six tutorials:

1. *Basic GRiNS/SMIL Editing & Uploading Skills*: a basic introduction to the GRiNS Editor, showing you how to take a simple template and fill in data objects, plus an overview of what you need to do to get the world using your productions once you've made them.
2. *Extending an Existing Presentation*: an overview of how slightly more complex presentations are structured and an overview of basic timing control within a GRiNS presentation.
3. *Working with Templates, SMIL Regions and GRiNS Resource Channels*: a description of how you can use GRiNS to build your own templates and on managing presentation resources (such as screen and audio spaces).
4. *Using Transitions and RealMedia*: an overview of the facilities in GRiNS for using RealMedia data types, such as RealPix, RealAudio, RealVideo, RealText, and RealFlash.
5. *Building Adaptive Presentations*: an overview of the facilities in GRiNS for helping you quickly build presentations that can adapt to the environments that your users may encounter.
6. *Hypermedia Support in GRiNS*: an overview of using the hypermedia support facilities available in GRiNS for links within objects, inside of a single SMIL document and across SMIL documents.

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

The first tutorial is intended to make sure that all of the components of your system are working correctly. We supply a sample document, and let you go thru the process of creating a document for use with the GRiNS Player. All other tutorials will refer to this section as a guide to the last steps in getting your presentation on the Web.

The GRiNS/SMIL team is constantly producing new documents to make using all of the features of GRiNS/SMIL as easy as possible. Check the GRiNS/SMIL Web site for updates to our documentation regularly at: <http://www.oratrix.com/GRiNS>.

## Introduction

This guide will help you to learn how to create SMIL documents using GRiNS that are optimized for use with generic SMIL-compliant players.

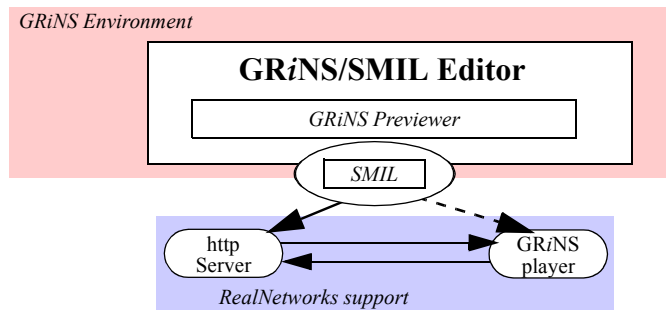
SMIL is a multimedia description language that was defined (and is maintained and further developed) by W3C: the World-Wide Web Consortium. GRiNS was used to help W3C debug the original SMIL standard, and the GRiNS Editor is the only visual authoring environment available that gives you access to the full power of SMIL.

Often, SMIL documents created using GRiNS/SMIL will be delivered to users across the Internet via the RealSystem G2 streaming multimedia server. If you wish to make use of the special facilities available via the RealSystem, you should consider using the GRiNS for RealSystem G2 product suite instead of the GRiNS for SMIL products. While GRiNS/SMIL provides basic support for RealMedia data types, the GRiNS/G2 product suite provides custom tools for making G2 presentations.

You can also create local presentations using GRiNS/SMIL — that is, presentations that do not need to be accessed via a streaming server — that can be played back directly from your local disk using the GRiNS/SMIL player. Local presentations are useful if all your users are on a single high-speed network, or if you don't expect wide distribution of your work. Local presentations are also useful if you are planning to distribute your presentation on a CD-ROM.

The GRiNS Editor uses an embedded previewer that helps you incrementally look at your presentation as it is being created. GRiNS also allows you to open and edit existing SMIL documents (whether initially created with GRiNS or not) and fine-tune them for playback via a SMIL-compliant viewer.

The following illustration gives an overview of GRiNS/SMIL:



## Things to Keep in Mind

Creating a multimedia media project is not a trivial task. It requires careful planning and the ability to incrementally craft your final message. Of course, you don't *have* to spend that much time planning: you can grab a couple of images and a bit of music and pound out a few lines of text, but unless you are this generation's Mozart — a musical magician who got things right the first time, every time — a bit of time at the drawing board can save you time, effort and money in the long run.

Before you create a document, you should first develop a clear understanding of your target audience. If your audience is spread across the Internet, remember that different users expect different experiences from your presentation. If your viewers are connected to the Internet via a high-speed link (such as a cable modem or a DSL line), they probably will not be easily impressed by static slide shows and fixed-format presentations. If, on the other hand, they are still connected using the lower speed modems, they'll be happy with any visual effects you can give them — as long as it doesn't take too long to download! In all cases, remember that end-to-end performance across the Internet is unpredictable and that careful resource use during presentation creation can make all the difference when the streaming server is out fighting to make sure all of your bits arrive on time.

## The GRiNS Workflow

GRiNS/SMIL was developed to help you manage the tasks of creating compelling presentations. The typical GRiNS workflow is:

1. create or assemble the media objects that you plan to use;
2. create a mental model of your presentation, including general story structure and graphical layout of where things will be placed on the screen;
3. incrementally create and preview the presentation using the various GRiNS *views* — each view has been designed to help you solve particular presentation problems;
4. add optional content that can be users with very fancy (or very vanilla) systems, or for users in particular target groups;
5. export your presentation for playback via a SMIL player — that is:
  - use the GRiNS bandwidth analyzer to make sure your presentation ‘fits’ on the network,
  - have GRiNS upload your presentation to a Web server

The actual steps you take will depend on the resources you have available when you create your presentation. GRiNS provides the following features to make your authoring activity more productive:

- GRiNS allows you to open any existing SMIL document — even if it was created using a Brand-X tool — and edit it using GRiNS. This not only helps you maintain presentations created by recently-departed staff members, it also lets you build your own templates with a particular look-and-feel.
- GRiNS lets you work with the content editors that you already use. If you select Edit Content on a media object, GRiNS will activate whatever tools you’ve configured on your system without having to leave the Editor.
- GRiNS gives you a ‘start anywhere, see anything’ model of previewing your presentation. You can preview a single media object, a group of objects or the whole presentation — no more running the entire show to see the last 10 seconds!
- GRiNS provides embedded bandwidth checking and analysis to help you manage your presentation resources. We do this automatically for you before you upload the final version, but you can also ask for a check at anytime during development.

- GRiNS allows you to target your presentation to a wide audience, without alienating people on the fringes: you can create your mainstream presentation first and then incrementally add support for the high- and low-end users, or for people who speak different languages or who prefer, say, audio above visual information.
- GRiNS brings the power of hyperlinks to streaming media. Let's face it, the Web is interesting not because of the content on any one page, but because you can easily link all sorts of information together. GRiNS gives you this same power for multimedia: you can easily add hyperlinks to either navigate within a presentation or to give the user the ability to jump from one presentation to another. Try it, you'll love it!

Of course, GRiNS also offers you other advantages. Chief among these is the GRiNS/SMIL structure-based approach to presentation creation. But: rather than talking about how GRiNS can help you, this guide will show you how to make presentations using many of GRiNS's facilities.

## **A Tutorial Introduction to GRiNS**

This guide will walk you through the creation of a complex presentation. We will create the presentation in steps, each of which are intended to highlight either basic GRiNS editing techniques or special GRiNS features.

The basic premise of all of the tutorials is the creation of a short walk through a city. We'll look at the issues involved in creating all aspects of this presentation.

This guide can be used either as a multi-step tutorial or as a GRiNS user's guide.

## **A Note on the Illustrations and Conventions Used in this Guide**

GRiNS/SMIL is available for Windows, Macintosh and Linux platforms. This section discusses the impact that platform dependencies have on the content of the tutorials. Most of the dialogs and menus are identical on all three platforms, so this guide can be used for all three platforms. However, users of the Mac and Linux versions should keep the differences summarized below in mind.

### *Microsoft Windows*

In order to provide a readable tutorial set, this Tutorials Guide provides illustrations and examples drawn only from the Windows version. GRiNS adheres to most of the standard conventions for Windows software, except as noted in the text below. Please consult the GRiNS *Release Notes* for the most recent updates on product performance and conformance.

### *Apple Macintosh*

Since the Mac has only a single-button mouse, you should substitute a CTRL-click selection where the text refers to a selection using the right mouse button. Contextual menus (selected with right-mouse on windows) will pop up on the Macintosh if you keep the mouse button depressed for about half a second.

The Macintosh does not have the concept of windows-within-windows, hence each open view has its own window. There is one point where this can be slightly confusing: it is possible to have a document open, but no views (open windows) on this document. The Window->Close directive closes a single view, and File->Close closes the document (and all views). The File->Open Documents menu shows the open documents and allows you to activate one of them.

The player toolbar is shown when you open the player view; there is no Editor toolbar. The keyboard shortcuts follow standard Macintosh practices.

### *Linux, SGI and SUN UNIX notes*

Each view in the Editor under UNIX has its own window. Moreover, each window has a private menubar. The order of menus and commands within menus is the same as on Windows and Macintosh, but menus and commands that have no meaning within a certain view are omitted. UNIX keyboard shortcuts follow standard UNIX conventions, and are listed in the menus.

Under UNIX, GRiNS uses a small separate window where you can open documents and quit the Editor, and another small window per open document where you select which views to open, save the document, etc.





# Tutorial 1: Basic GRiNS/SMIL Editing & Uploading Skills

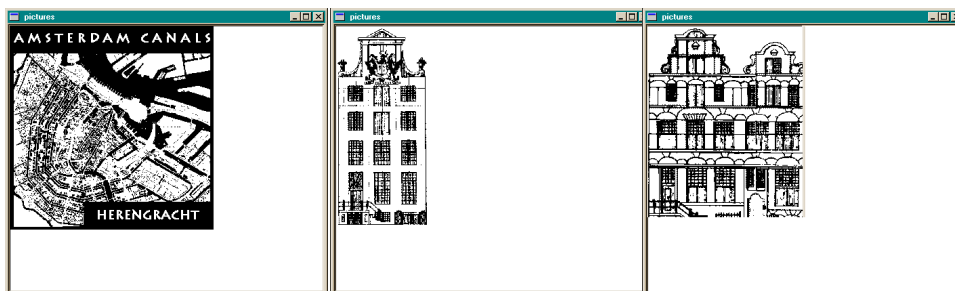
## Overview and Goals

The purpose of this tutorial is to help you perform confidence checks on the GRiNS/SMIL environment and to help you determine the special parameters and settings you will need to upload your presentations to a Web Server.

In order to create this small presentation, you will perform the following steps:

1. Start the Editor and select the *QSG1-Slideshow* template;
2. Go into the Structure View;
3. Select each object in the template and define a image to be associated with each block;
4. Preview the presentation
5. Publish the presentation.

The result of building the first part of this tutorial is a series of three pictures displayed in the (player-dependent) default location on the screen. A thumbnail presentation preview is:



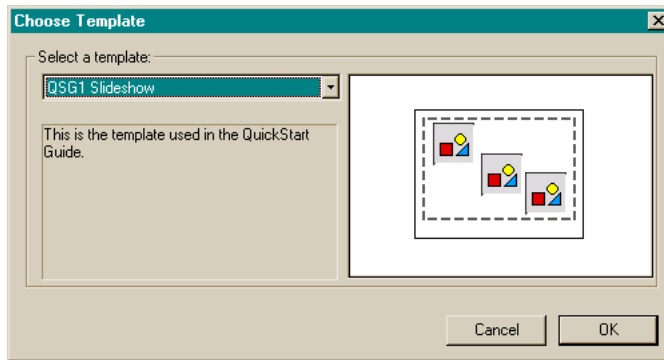
We close this tutorial with background information on GRiNS that will make your creation of template-based more effective.

### NOTE:

1. A simplified version of this tutorial was provided with the Quick Start Guide. This version contains extra information, and we recommend you go thru the steps here again.

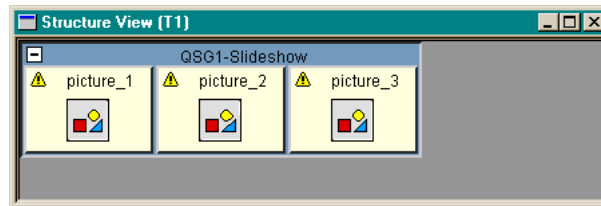
## Opening a Template File

If you have not already done so, start-up the GRiNS Editor; if you get a pop-up box asking if you want to create a new presentation or edit an existing one, select New Presentation. Otherwise, if no pop-up appears, go to the File menu and select New, which allows you to select a GRiNS Template file. Select the *QSG1-Slideshow* file, as shown below:



This will create a document named *Untitled1.grins*. Before continuing, save your work-in-progress as *T1.grins* in a directory on your system. Do this by selecting **Save as ...** from the File menu.

Whenever you create or open a file, you see the Editor's Structure view. In this view, we see you blue outer container labelled *QSG1-Slideshow* and a series of three light-colored nodes called *picture\_1*, *picture\_2*, and *picture\_3*. (The warning triangles indicate that the nodes are empty.)



At the top-left of the structure container is an expand box. Give it a click to see one of GRiNS/SMIL's information hiding features.

#### NOTES:

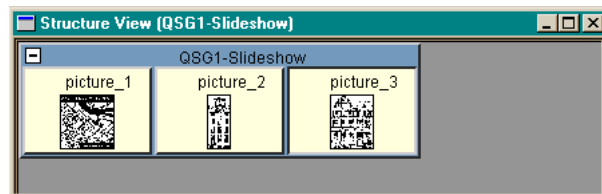
1. The opening screen assumes that you use the default configuration for GRiNS as distributed;
2. Some of the default dimensions of various GRiNS windows have been changed in the illustrations in these tutorials to make for a space-efficient presentation;
3. The colors used for the various GRiNS windows depend on the Windows color scheme installed for your system. This tutorial uses the Desert color scheme;
4. If you work on a multi-user system, make sure you don't overwrite files created by other users of this tutorial. On a multi-user system, each user should create their own copy of the Tutorials.

### Inserting the Media Objects

The *QSG1-SlideShow* template contains all of the information required to create a simple presentation *except* the actual images used. To get started quickly with GRiNS, do the following:

1. Using the Windows file exploring, open the *Assets* folder in the *Tutorials* directory (typically located in the GRiNS root directory; this is usually *c:/Program Files/Oratrix/GRiNS\_SV1*, which is the default location on Windows platforms);
2. Select the image *map.gif* from the folder and drag it on to the light-colored box labelled *Picture\_1*;
3. Select the image *h168.gif* and drag it on to the box labelled *Picture\_2*;
4. Select the image *h218.gif* and drag it on to the box labelled *Picture\_3*.

The illustration shows a 'filled' version of the structure view, containing thumbnail icons for each of the images you have placed on the presentation.



## Previewing the Presentation

### *Playing the entire presentation*

In the *QSG1-Slideshow* template, each object has a default duration of 5 seconds. As a result, when you play the presentation, you see each object for five seconds, with each new object replacing the old.

To see this behavior, select **Play**. You can do this in three ways:

1. using the triangular **Play** button on the shortcut bar;
2. by selecting **Play -> Play** command from the menu bar
3. by selecting the **Ctrl+P** key combination.

### *Playing part of a presentation*

GRiNS allows you to selectively play parts of a presentation.

To get a quick taste of what GRiNS/SMIL can do, select the box labelled *Picture\_1* (it contains the icon of the *map.jpg* image). Go to the **Play** menu and select **Play Node**. Once you do this, you get to see the Map image. Only that single node is played. You can use **Play Node** on any GRiNS node (data nodes or structure nodes).

You can also select **Play From Node** in the **Play** menu. For example, select the box labelled *Picture\_2* in the Structure View. You now see both the second and third pictures — that is, all nodes are played starting at the point that you selected **Play From Node**.

The ability to selectively activate and preview nodes or sub-parts of a presentation is a powerful tool that can help you fine-tune your presentation with a minimum of authoring effort.

## Publishing the Presentation

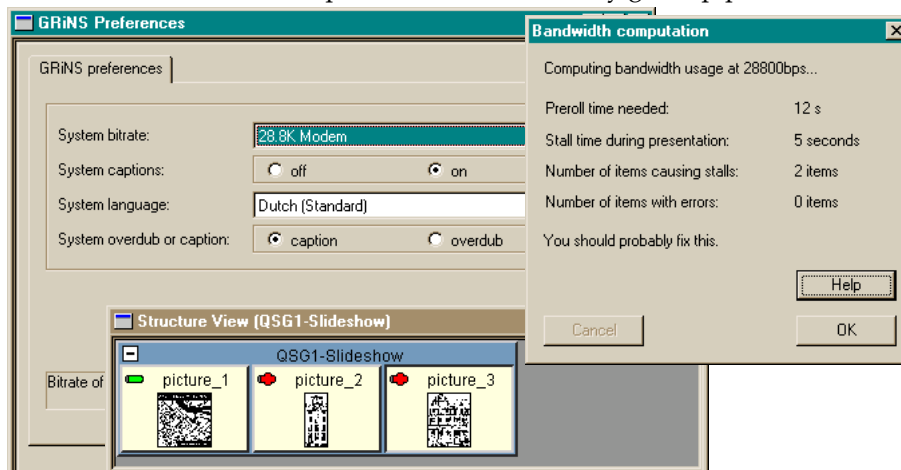
Publishing your presentation consists of several steps, some of which are optional.

### *Check Bandwidth Usage*

First, you should check to see if the presentation you've made 'fits thru the pipe' between the Server and your user. GRiNS provides several features to help you do this:

1. you should select a system bitrate from the **Edit->Preferences** menu.
2. next, select **View->Check bandwidth usage** to see if your presentation will fit thru a line with the properties defined in (1).

GRiNS does not correct errors for you (since such corrections involve artistic design issues), but it does indicate where you should look for the problem spots in your presentation. These are indicated by red blocked pipes on data items. Items which do not cause a problem are indicated by green pipes on data items.



The following situation shows the result of a bandwidth computation with settings for a 28.8 modem. The problems shown in this report can usually be fixed in one of several ways:

- by increasing the duration given to load the image (in this case, increase the duration of picture\_1 from 5 to 10 seconds); or
- by reducing the complexity of an individual data object (by using a different format or by cropping out unwanted part of the image).

### *Generate the Presentation*

The next step is to actually generate the presentation. GRiNS/SMIL does most of the work for you, which consists of:

1. creating a folder named *<filename>.data*, which will contain copies of all of the media objects in your presentation;
2. performing a bandwidth check, and
3. creating the actual SMIL file (*<filename>.smi*)

All of these steps are automated; they are triggered by the File->Publish command. Once the process is done, you will have the following items in the folder containing the *T1.grins* presentation:

- *T1.grins*: the GRiNS project file
- *T1.smi*: the SMIL version of the GRiNS file
- *T1.data*: the directory containing all data objects

#### Notes:

1. If you have RealSystem G2 installed, the icon associated with the *.smi* file will typically be that of Real's G2. (G2 'claims' the *.smi* file type, typically without asking... .) You can use this file with any SMIL player *except* G2, because of G2's limited tolerance for non-G2 data types.
2. The main difference between the *.grins* file and the *.smi* file is that all of the media object references have been localized in the *.smi* file. If you want to distribute your files to other users, make sure you give them the *.smi* version.

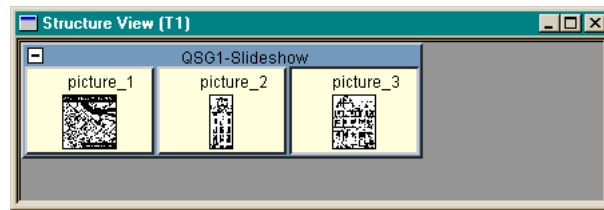
### *Generate the Presentation and Upload*

Once the presentation has been created, you can also upload it automatically to a Web Server. Select Edit->Publish and Upload to do this from GRiNS. Please check with your system or network administrators to set the various upload parameters in the Document Properties dialog; these are required for the Server you plan to use for hosting your presentation.

## Additional GRiNS Features

### *Understanding the Structure View*

The Structure View gives you an overview of the structure of any SMIL presentation. If we look at the contents of the presentation just created, the following detailed information is shown in the Structure view:



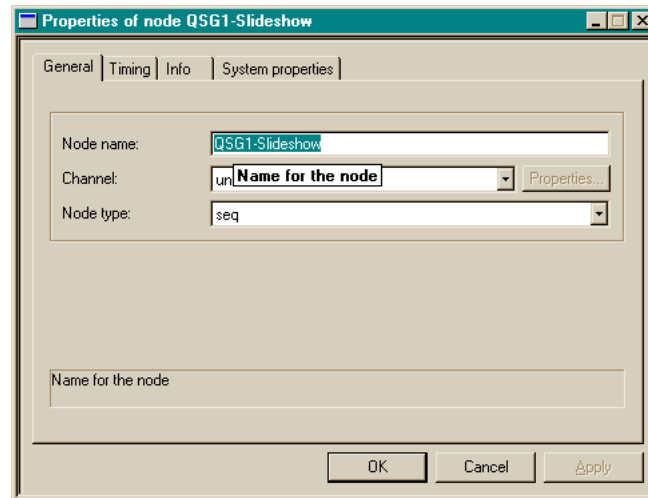
The presentation consists of three media object references (the light-colored boxes) and one structure node (the blue box labelled *Title*). Each of the media object references, labelled *picture\_1*, *picture\_2* and *picture\_3*, refers to a place where you can insert an image in to the presentation. An object reference describes all of the uses of a particular object that is stored either locally or at a remote site. If one image is used multiple times, each use will be described by one image object reference. Each reference will describe the duration of the image, its location on the screen and other properties associated with the use (or instance) of this image. All objects have defaults associated with the type of media being rendered.

It is important to understand that media object references (the light colored boxes) have attributes that control how the media objects themselves are played during the presentation. Each of the structure elements also have their own attributes that can be used to tailor a presentation.

### *Changing attributes of GRiNS Objects*

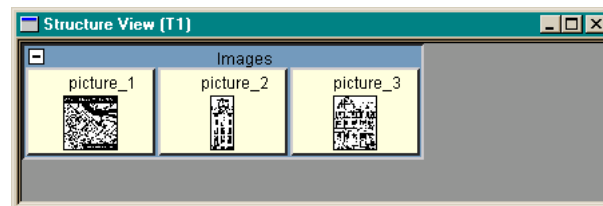
In order to get a feel for working with the attributes of objects, let's change the names of each of the objects shown in the presentation. This isn't strictly required, but can help make your SMIL code more readable by others — and it becomes important when you want to work with hyperlinks or if you want to do complex timing operations on your file.

Start by bringing forward the Structure view; if this view is obscured, you can always bring it forward via the Windows menu. Now, select the node named *QSG1-Slideshow* and bring up its Property dialog using either the right mouse button or via the Edit menu. The Property dialog contains high-level properties of the object:



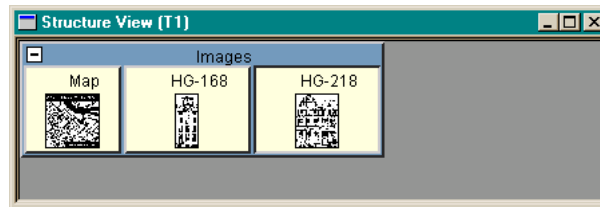
The overview Property dialog says: *Title* is a sequential node containing three children.

Change the name to *Images*, then click on OK. The resulting Structure view is shown, with the *Images* as the title of the structure node.





You can also rename the individual media objects to: *Map*, *HG-168* and *HG-218* by opening their respective Property dialogs. This yields the following Structure view:



## Closing Comments on Tutorial 1

The purpose of this tutorial was to acquaint you with the basic features of the GRiNS/SMIL Editor. In this tutorial we worked primarily with the Structure view, which is the main editing view in GRiNS. We also used the Player view, which contained the actual presentation contents. GRiNS/SMIL also supports other views of the presentation, such as the Timeline View and the Layout and Hyperlink views. You can always select the views you want open from the View menu; you can have as many open as you need, and you can move each view within the GRiNS overall canvas to help organize your editing session.

In the next tutorial, we'll revisit these features and show how you can also create more complex GRiNS/SMIL presentations easily by manipulating the structure of an application.

For now, take a break and give *Tutorial 2: Extending an Existing Presentation* a try when you are ready.



## Tutorial 2: Extending an Existing Presentation

### Overview and Goals

In this tutorial you will extend an existing SMIL presentation that displays a sequence of images in three separate screen regions by adding background music.

In order to edit this presentation, you:

1. start the GRiNS/SMIL Editor and open the *Walk.grins* presentation;
2. preview the presentation;
3. learn to navigate through the structure of the presentation;
4. add some new nodes containing background music;
5. fine-tune the timing relationships between objects using property dialogs;
6. further manipulate timing relationships using the GRiNS Timeline View; and
7. get a taste of manipulating the layout of the presentation via Layout Channels.

This tutorial assumes that you have the GRiNS Editor available, and that you have developed the basic skills illustrated in *Tutorial 1*. Some of that material will be reviewed here, but your time can be most productively spent if you look at *Tutorial 1* before you complete this one.

The following sequence of images shows what the first part of the tutorial will yield. (Note: the audio is not shown.) Each of the images is bordered by a box showing the boundaries of the respective regions used. These are shown for illustrative purposes only, and are not shown in the actual presentation.

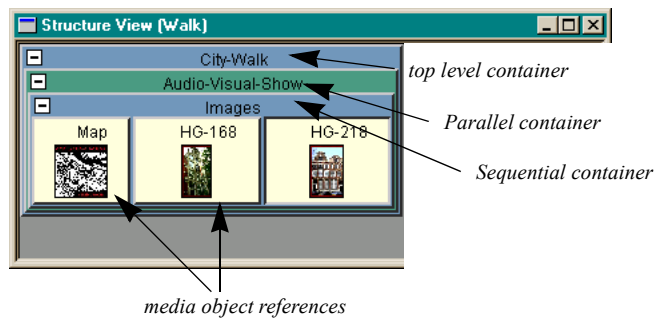


## Creating the Presentation

In the first tutorial you opened a standard GRiNS template. In this tutorial, you open an existing presentation and expand it with new content. You can do this with any legal SMIL presentation — whether created by hand with a text editor or with someone else's editing product.

### *Open the Walk.grins presentation*

Start-up the GRiNS Editor and open the *Walk.grins* presentation by selecting **Open** from the **File** menu, and browsing the *Tutorial* folder in the GRiNS system folder. When the document opens, the **Structure View** is activated. This will give you the following Editor view:



The structure in this view is the same as that in the first tutorial, except that:

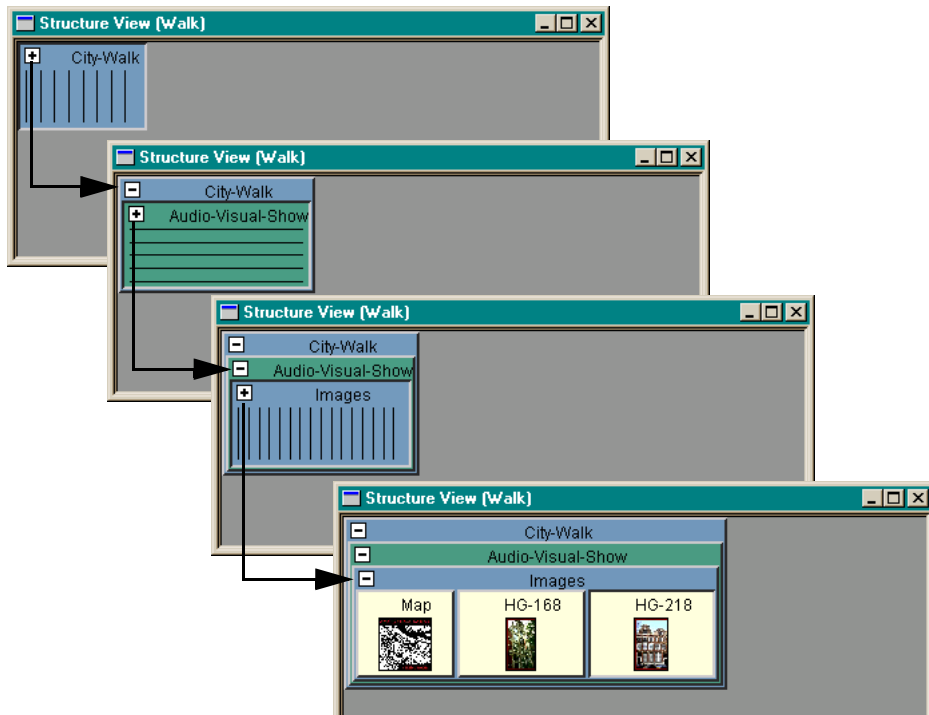
- the media objects are slightly different (they are color jpeg images)
- there are additional structure containers (*City-Walk* and *Audio-Visual-Show*.)

Once you open this presentation, you can edit and save it under its existing name or you can save it under another name (such as *Walk2.grins*). If you save it in another directory, make sure you also retain access to the *Assets* folder, since the presentation uses its path name for media objects.

### Revisiting the GRiNS Structure View

In the first tutorial, you saw that a blue structure box with a stack of three items was the GRiNS method of representing a SMIL sequential structure container (SEQ). In this document, you see the same structure in the container named *Images*. There is also an additional green structure container outside: a parallel (PAR) container labeled *Audio/Visual Show*. There is also an outer blue sequential container named *City Walk*.

In order to see how GRiNS can help you manage the complexity of the Structure view, click on the small triangle at the upper left corner of the *Images* node. This has the effect of collapsing the details of the node, leaving you with only a “closed” structure container. The results of closing the containers available are:



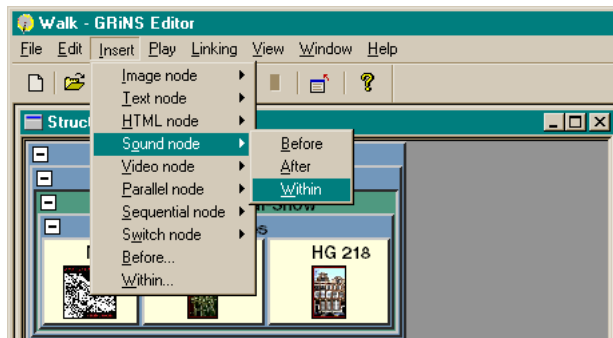
In this simple presentation, opening and closing boxes is not necessary, but this facility is really handy once you start to edit more complex SMIL presentations. But: enough about the future — let's get some work done!

The first thing you should do is to play the existing presentation. You can do this by selecting the VCR-like Play button on the tools bar or by selecting Play from the Play menu. You can also play individual nodes with Play Node or Play From Node.

### *Adding a new media object reference*

A presentation like this short walking tour of a city is nice, but you can liven things up a bit by adding some music. In order to do this, you first need to tell the Editor that you want to define a new node in the presentation, and then fill the node with a particular media reference.

To create a new node, make sure the target structure container is open (by selecting the appropriate triangle), which in our case is the green node labelled *Audio-Visual-Presentation*. Now create a new node by going to the Insert menu and selecting Sound Node -> Within, as shown below:

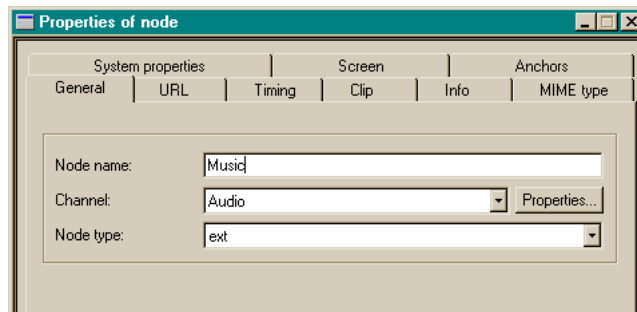


This operation says: in the currently selected structure node (in this case, a parallel node), add a new sound object within the node. We could have also placed the new node before or after the structure node by using the appropriate commands.

Whenever you create a new node, a *node creation dialog* pops up, in which the basic properties of this node can be defined in the node's Property dialog. Fill in the following properties, as shown below:

- *Node Name*: Music;

The *Node Type* is automatically set to *ext* (for external object reference) and the *Channel* is set to *Audio*.



You can now associate a particular media object with this node by Drag and Drop, or you can use **Browse** in the URL tab to select the name of the media object. Since we've already used GRiNS drag-and-drop, try **Browse** and locate the file *tune.aiff* in the *Assets* folder. The result is:



Before we preview the result, let's review what we have actually done in terms of the underlying SMIL file.

The structure view shown above says: the presentation consists of one parallel node that contains two elements: a piece of background music and a sequential structure element that itself contains three images.

Once you get used to this representation, you can immediately see what happens in parallel and what happens sequentially in a presentation. The Structure View given in the presentation corresponds to the following SMIL structure:

```
<seq id="City Walk" >
  <par id="Audio-Visual-Presentation" >
    <seq id="Images" >
      <img id="map" ... />
      <img id="HG_168" ... />
      <img id="HG_218" ... />
    </seq>
    <audio id="music" ... />
  </par>
</seq>
```

Spend a few moments to understand how the SMIL code and the Structure View are related. Once you understand this, using GRiNS to make complex SMIL presentations is a snap!

### *Manipulating Structure and Data Objects*

GRiNS allows you to manipulate both structure elements in a presentation and individual data objects. Why? GRiNS does this because being able to manipulate structure makes presentations easy to expand, edit and maintain. By separating structure from content, you can reuse the same structure dozens of times. We can also substitute and update the individual data objects without changing the base presentation. We can also copy and paste the structure from one presentation to another, something you'll see in a later tutorial.



### *Previewing the Results*

The presentation will display a sequence of three images (for 5 seconds each) in parallel with a short background audio track. You can see and hear the result by selecting **Play** presentation from the **Play** menu or by using the VCR-like icon on the shortcut bar. Try this now.

If you used our data objects, you will see three images in sequence (each displayed for 5 seconds, for a total of 15 seconds) displayed at top-left, middle and bottom-right on your screen, and you hear a bit of music in parallel that lasts for about 4 seconds. Having the music be this much shorter than the full presentation is not optimal, but it gives us something to fix later in this tutorial!

While you have just previewed the entire presentation, the GRiNS Editor also allows you to preview only parts of the presentation. For example, select the audio node, and then select **Play** node from the **Play** menu or with the right-mouse button. You now only hear the audio. You can do this for any simple data node, but you can also do it for any structure node. For example, select the structure node labeled **Image Sequence** once, and then select **Play** node from the **Play** menu. All of the images in the sequence (that is, the contents of the selected node) will be previewed, but the accompanying audio — which was not selected — will not.

#### **Note**

1. Be aware that the Player window will be pushed to the background if the Structure View is accessed; you should get into the habit of moving the Structure View a bit to the right to see if the Player is being blocked.

Congratulations: you've made your second GRiNS-based SMIL presentation.

## Fine-Tuning the Timing of a Presentation

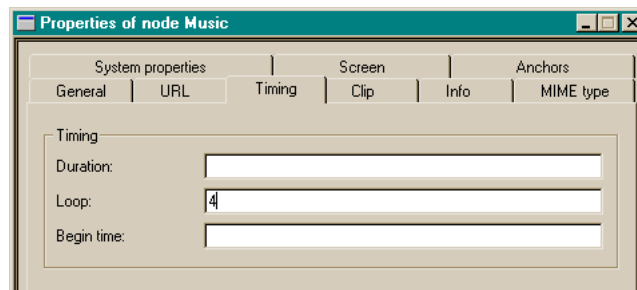
In this section, we show five simple editing operations that will start to develop your SMIL skills for extending the control among objects in a presentation:

1. you can tell GRiNS to loop your data object for a certain number of iterations;
2. for objects with a behavior like background music, you can set the loop to be determined by its context in the presentation;
3. in a parallel group, you can explicitly say which object controls the end of all of the members of the group;
4. you can use synchronization arcs in the Timeline view to control relative start times on objects in a structure container, and
5. you can change the assignment of objects to Layout Channels, and resize or move the Layout Channels to another area of the screen.

Each of these are discussed in the following sub-sections.

### *Setting an Object Reference to “Loop”*

Open the Structure View and select the *Music* data reference by clicking on it once. Using either the right mouse button, or the Edit menu, select Properties, which brings up the properties window for the audio node:



Using either the arrow keys or the mouse, select the line labeled Loop. In the attribute editing area at the bottom of the box, set the value to 4. This tells the player to loop the audio four times during this part of the presentation.

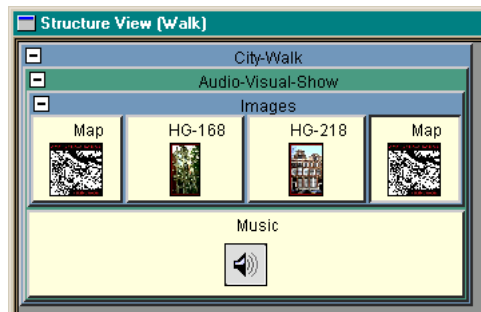
Now, preview the new presentation by selecting **Play** from the shortcut bar, the right mouse or the CTRL-P shortcut. The audio will keep repeating until all of the images have been shown -- a total of 16 seconds.

**GRiNS Tip:**

*Not all data types can be set to loop. For images or text (or any object that is discrete — that is, that has no 'natural' duration), looping does not take place in GRiNS. For all continuous, time-based objects, looping makes sense, but not all players support looping functionality. For streaming media (such as RealAudio), looping is not supported, since it requires an entire stream to be buffered in the Player.*

### Setting an Indefinite looping period

SMIL gives you lots of flexibility in controlling the duration of nodes, but using the kind of explicit control shown above can be dangerous if you want to reuse the structure of a presentation. For example, assume that you defined a Loop counter of 4 for an effective object length of 16 seconds.



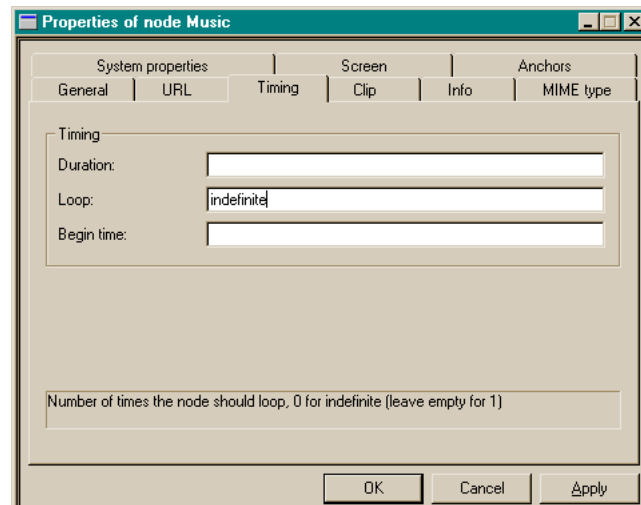
Now, edit the presentation as follows:

1. select the node labelled *Map* in the *Images* container
2. using either **Edit -> Copy**, the **Copy** function under the right mouse or CTRL-C, copy the *Map* element.
3. select the last image in the three-image sequence (labelled HG-218), and do a **Paste** from the **Edit** menu, via the right mouse or using CTRL-V.

This makes a copy of the node *Map*, so that the presentation now looks like the image above.

If you now preview the presentation, you get a 20 second image show, but only 16 seconds of audio — not exactly what you would like!

You can solve this type of problem easily with GRiNS. First, open the Properties dialog for the *Music* object. This is shown below.



Now, replace the loop count of 4 with the word *indefinite*, as shown. What this says is: keep looping the object until the end of the object's parent container. The duration of the parent is determined by the length of all of the images, so the music loops until all the peer-level objects have been completed.

It is important to realize that *indefinite* is not the same as *forever*: the duration of an object with indefinite looping is determined by the object's context. When the context ends, the object ends.

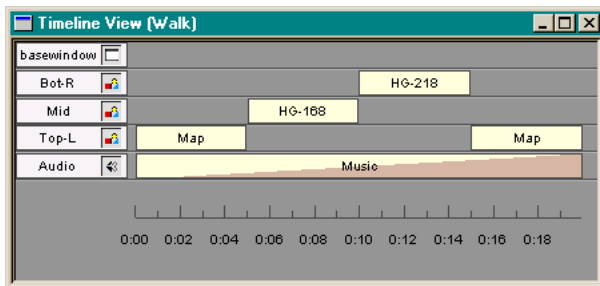
There are several other ways of influencing the timing of a node or of a structure container. We consider these later in this tutorial.

## Advanced GRiNS Features

This section presents some of the more advanced features of GRiNS/SMIL.

### *Introduction to the Timeline View*

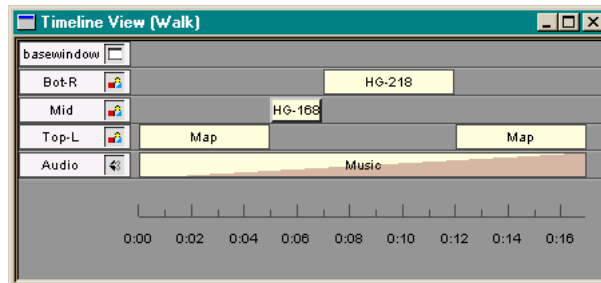
The Structure view shows the logical grouping of elements in a presentation. At runtime, the structural view is mapped to a presentation timeline. An indication of this timeline can be seen by looking at the Timeline View of a presentation. You can bring up this view by selecting it from the Windows menu. A picture of the timeline for this presentation is:

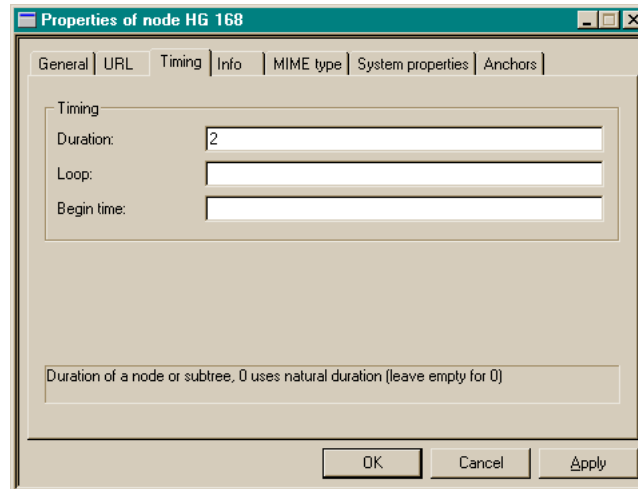


This view shows the *Music* object and *Map*, *HG-168*, *HG-218* and *Map*. Now, select *HG-168* and bring up its Properties via the right mouse button. Find the *Duration* property and change it from 5 to 2 seconds.

The Properties dialog for this node (shown on the next page) contains all of the SMIL and GRiNS-related properties associated with this node. The use of most of these properties will be covered in other tutorials.

The result of this is that the length of the *HG-168* and *Music* objects are scaled automatically. The runtime behavior can be confirmed by previewing the node or the entire presentation using the Play options.





Notes:

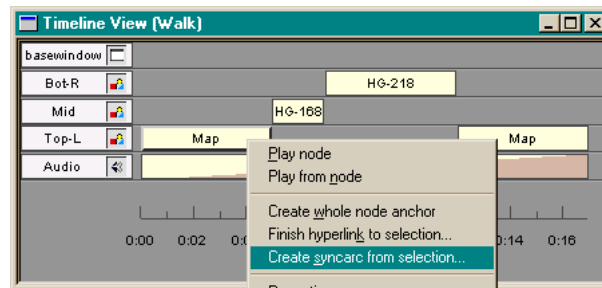
1. The **Timeline** displays either the natural or effective duration of objects. In the example above, the effective length is shown, which is the length provided by the runtime environment. For some datatypes (where information on duration cannot be drawn from the definition), the natural length or a best-estimate is shown.

### *Fine-Tuning via Synchronization Arcs (Sync Arcs)*

Let's suppose that you wanted to take our presentation-under-construction and make some fine-grained adjustments to timing. For example, let us assume that you want the audio fragment to begin slightly later than the first image. Perhaps you are doing this to balance the delivery load of the presentation, or maybe you just want to create a particular effect.

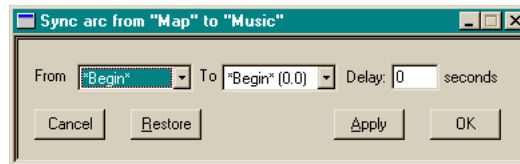
GRiNS allows for this fine-tuning via *synchronization arcs*, or *sync arcs*. Sync arcs provide a visual representation of special relationships between objects in the timeline. They are only available in the timeline, since they have to do with the execution state of a presentation rather than its logical state.

To create a sync arc, select a node in the Timeline view. For example, select the picture of the map (named Map) and then, using either the right mouse menu or the Linking menu, select the Create sync arc from selection command:

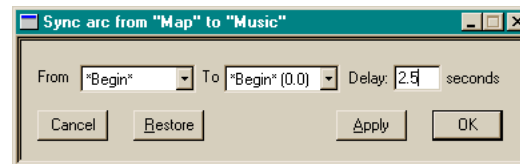


This will define the source part of a sync arc. The source is colored light green to highlight it in the Timeline view.

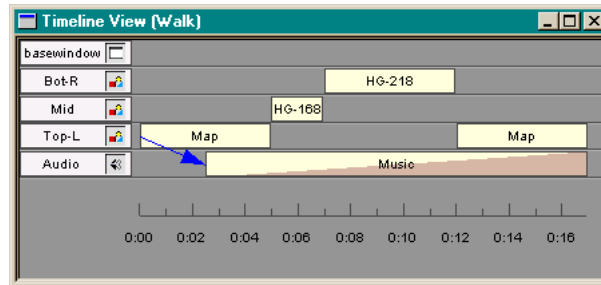
Once the amber node is shown, select the node named Music. This will bring up the Sync arc properties box, as shown below.



The GRiNS default is to define a relationship between the end of the source to the beginning of the target. In our example, change *From* to *Begin* and give a delay of 2.5 seconds, as shown bottom right.



When the sync arc is accepted, the arc is drawn as a blue arrow between the start of *Map* and the start of *Music*. It says that the audio node will start 2.5 seconds after the start of *Map*:



Once a sync arc is created, you can edit its properties by selecting the arrow head associated with the arc. If you make changes, the timeline will be redrawn automatically for you.

Notes:

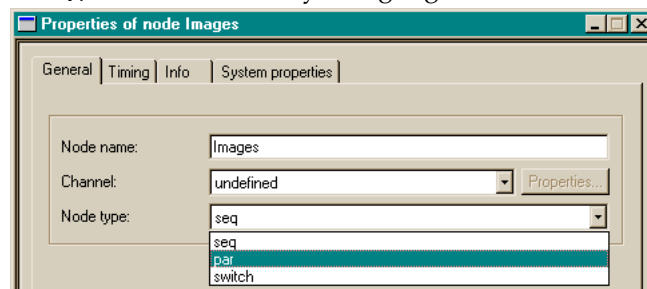
1. SMIL places structural restrictions on the scope of sync arcs. In SMIL V1.0, both the source and target must have a sibling relationship. This is expected to be relaxed in version 2.0.



### *More on Timing in Parallel and Sequential Structure Nodes*

The examples in this tutorial have shown you how to add nodes and adjust timing in a presentation. We conclude with looking at manipulations on SMIL structure nodes.

Go to the Structure View, remove the second copy of *Map* from the presentation (with Edit->Delete, CTRL-DEL or the Delete popup menu entry). Next, select the *Images* structure node and bring up its Property box. In the middle part of the top of this box is the Type selection box. If you highlight the alternatives, you get:



If you select *par* (as shown above) instead of the original *seq*, GRiNS will take the image sequence and turn it into a parallel structure group in which all of the images are displayed simultaneously.

If you confirm the selection of *par* by selecting Apply and then preview the presentation, you will see that the images are played together. The images *Map* and *HG-218* are displayed for five seconds each, and *HG-168* for two seconds. The audio lasts as long as the longest image (instead of the sum of the image times).

A copy of the screen at the start of the presentation is shown on the next page.



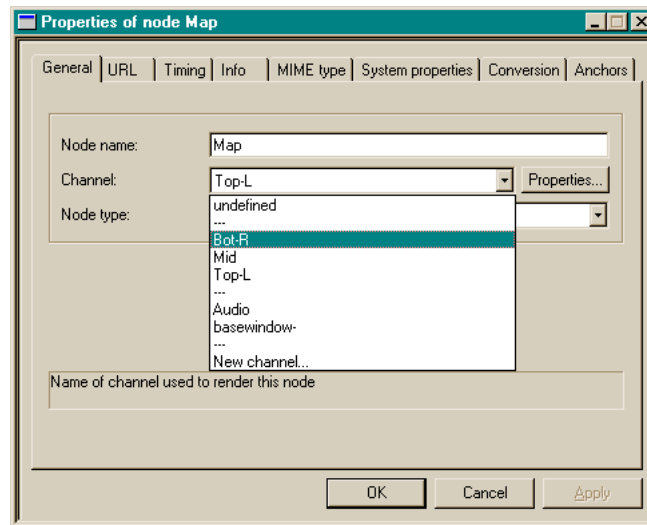
The presentation as it now stands has a delay defined for the start of the audio. This means that the *Map* image is shown for 5 seconds, but *Music* starts 2.5 seconds after the *Map* has been shown.

### *Selecting Layout Regions*

All of the examples you have used in this tutorial play their output on to a **Resource Channel**, which is the GRiNS way of identifying which screen and audio resources are used to render the presentation. In the first tutorial, you used the default resources available, as defined in the SMIL standard. In the second tutorial, you used four **Resource Channels** that were defined as part of the *Walk.smil* presentation.

In the next tutorial, we take a deeper look at Resource Channels. To get a taste of this kind of presentation control, we end this tutorial with a simple introduction on how to associate any data object with its presentation space.

Open the Structure View and select the *Audio-Visual-Show* object. Go to its Property dialog and make sure that the object type is *par* (for parallel node). Confirm this choice with OK and then select first element in the *Images* — this is the *Map* object. Next, bring up the Properties Dialog for this image by selecting Info from the Edit menu (or from under the right mouse button). In the upper right hand corner of the dialog is the Channel selector. When the list of available channels is exposed, you will see the following options:



If you select Bot-R, the *Map* image will be displayed at the bottom-right part of the screen. Select Bot-R, confirm the selection and then preview the presentation using PLAY. If you also change the second image (*HG-168*) to display its output in Bot-R, all three images will be displayed in the bottom right part of the display.

## **Closing Comments on Tutorial 2**

The purpose of this tutorial was to acquaint you with the notion of parallelism in a SMIL presentation and to let you get an idea of SMIL abilities to manage the flavor of a presentation by using looping and various forms of activation and duration control.

Unlike the first tutorial, this one also made use of the ability to explicitly place objects using GRiNS Resource Channels. The next tutorial looks more closely at the types of control you have via GRiNS layout.

For now, take a break and give Tutorial 3: *Working with Templates, SMIL Regions and GRiNS Resource Channels* a try when you are ready.

## Tutorial 3: Creating Templates, Working With Regions and Using GRiNS Resource Channels

### Overview and Goals

The purpose of this tutorial is to create the *Walk.grins* SMIL presentation that was used in Tutorial 2. The *Walk.grins* presentation contains references to four media objects and it makes use of three visual rendering areas on the screen. It also makes use of one audio rendering area. Since Tutorials 1 and 2 show how to define and integrate media objects into a presentation, this tutorial will concentrate on defining SMIL Regions by using GRiNS Resource Channels.

You will perform the following steps in this tutorial:

1. Open the Editor and select the *Empty* template;
2. Define three Resource Channels, one for each of the visual regions;
3. Define the structure and media object referenced to be used in the template and define a Resource Channel for the audio object;
4. Assign default timing to nodes where appropriate;
5. Save the presentation for later use.

Once the presentation has been saved, it can be imported as a template.

### An Introduction to Layout Manipulation in GRiNS and SMIL

GRiNS helps you allocate and organize presentation resources by using a tool called a Resource Channel. A Resource Channel is a place where you can put information that is all managed using a common policy. Often, the Resource Channels will help you manage space on the display screen. In more complex presentations, Resource Channels may help you control an audio device (or one channel on a stereo audio device), or a printer, or a control/communication program or a special-purpose device.

In most initial examples of GRiNS/SMIL, it is convenient to think of a Resource Channel being the same as a SMIL *Region*. A SMIL *Region* is a rectangular area of the screen. It is defined by an X, Y base coordinate, a height and width (in pixels

or as percentages of the available space) and a ‘depth’ coordinate (called Z) which tells you how individual objects are ordered on top of one another.

Where the SMIL *Region* provides a means to partition the presentation canvas, the Resource Channel in GRiNS provides a broader perspective on managing presentation resources. Using GRiNS Resource Channels, you can give individual collections of data objects a common resource control policy, which can help the Player decide how best to process your documents.

A general discussion of Resource Channels is beyond the scope of these tutorials. In this tutorial, we concentrate on using Resource Channels to manage the SMIL-defined properties associated with *Regions*. To simplify the text, we will use the name Channel to denote a GRiNS Resource Channel and a Region to denote the properties specific to a SMIL *Region*.

### *Layout and the Walk.grins presentation*

In Tutorial 2, we made use of three SMIL regions, one of each picture. In this tutorial, we show you how to define and manipulate these regions, and how to customize them for special needs.

When you previewed the *Walk.grins* presentation, you saw that each of the three images were placed in different parts of the screen: the *Map* was placed in the upper left part of the display, the first house near the middle, and the second house in the bottom right. Each of these areas corresponds to one Channel. There was also a fourth Channel in the presentation: the background area that defines the size of the Player window. The fifth resource used in the presentation was the audio device — this is a device that is not specifically controlled in SMIL presentations, but one you can influence when you use GRiNS.

In *Walk.grins*, each media object is placed on a separate channel. This is acceptable for small presentations, but it is typically not a good idea for larger presentations. GRiNS provides facilities for helping manage all the resources in the presentation, but you can help GRiNS by reusing resources when appropriate. You will see how to do this in the following examples.

In conventional HTML presentations, the user is not given very much control over the visual area in the presentation. This makes a lot of sense for the text-flow documents that make up the majority of HTML presentations, but it is very frustrating to graphic designers and media presentation creators, who often want much greater control over the relative placement of objects on the display. Although several mechanisms exist in HTML to gain more control over the display space, using these often require a full implementation of CSS-2 (the optional Cascading Style Sheet extensions to HTML browsers). In order to provide a basic degree of control over SMIL presentations, the creators of SMIL decided to provide a simple, light-weight layout mechanism that any SMIL Player could easily implement. This mechanism is called *SMIL Basic Layout*.

*SMIL Basic Layout* allows you to define a base window in which your presentation is played. The size of the base window can be defined in terms of pixels or in terms of external unit measure (typically millimeters). Each of the SMIL Regions (if they are used) are placed inside the presentation's base window. Each Region is defined by X, Y coordinates that define the location of the Region within the base window, followed by either an absolute size (in pixels or mm) or a relative size (as a percentage of the base window).

When you define a presentation (or a presentation template), you specify how large a base window you want, and then how many independent regions you want to define to segment the space inside the base window. Of course, you don't need to specify any base windows or regions at all: in this case, you make use of the default layout properties of the Player you are using.

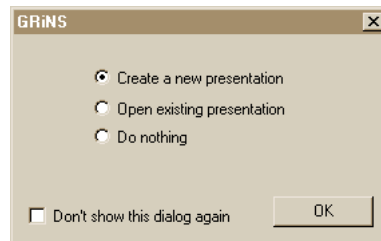
You specify a SMIL Base Window and the individual SMIL Regions via (Resource) Channels in GRiNS. The Channels provide a container for basic layout attributes and they provide an extra measure of control by allowing you to assign media types to individual SMIL Regions.

The GRiNS Editor also allows you to define *Layout Screens* (or simply *Screens*); Screens let you group a number of Channels that you would like to have available during a particular part of a presentation. This allows you to change the look of your presentation over time -- something that most viewers will appreciate.

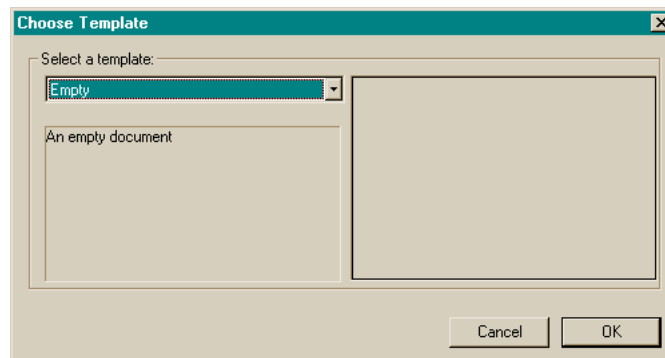
In this simple presentation, we will ignore the concept of a **Screen**. Instead, we make the decision to define three regions on the screen for displaying three images in a sequence in different places.

## Creating the Presentation

If you have not already done so, start-up the GRiNS/SMIL Editor. You will usually get the following pop-up box (if not, go to the File Menu and select New).

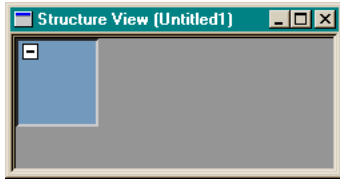


You will be asked to select a template file, as shown below:



Select the template named *Empty*. The *Empty.grins* template contains the SMIL boiler-plate information required to build a SMIL-based presentation.





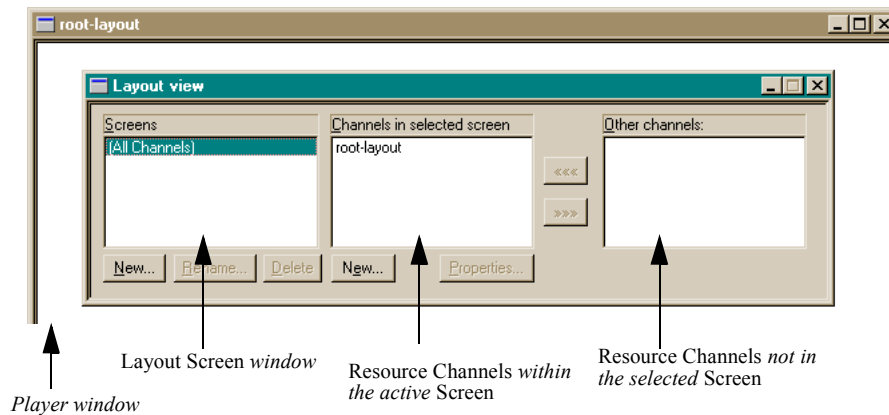
When you open the *Empty.grins* document, you get the default GRiNS Structure view, which in the empty document contains a single empty, unnamed sequential structure container.

Once the *Empty.grins* template is opened, you can start with the process of designing your application (or application template). There are many ways of building a SMIL presentation. You can take a *data-centric* view by first defining all of the objects in the presentation and their relative order or activation, or you can take a *presentation-centric* view by first designing your layout region(s), and then later assigning objects to each of these regions. In most design cases, you may switch between the two approaches — first defining a few regions, then defining media objects, and then adding some more regions.

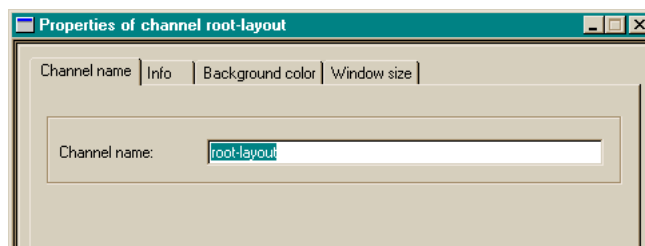
For the sake of simplicity, this tutorial takes the latter view: we start with designing a simple layout space and then identify the objects to be presented.

### *Define Regions for Displaying Presentation Components*

In order to define and manage your presentation resources, you should open the Layout View from the Windows menu. The Layout view looks like this:



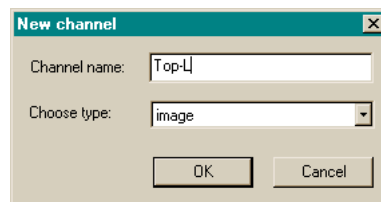
GRiNS provides you with a *root-layout*, which is the rendering canvas for the presentation. The default size of this canvas is 640 x 480 pixels. You can resize the window by editing the various properties associated with the *root-layout*; select the name *root-layout* in the middle (Channel) window of the Layout View, and then selecting the Properties button. This brings up the following Properties dialog for the *root-layout*:



The main elements to alter are the name, the background color of the window and the window sizes. Go to the **Background Color** window, and set the background to black. (You can either use the color editor or simply type in the word *black*.) We don't need to alter the other defaults now, so close the window to expose the Layout view window again.

#### *Define SMIL Regions via GRiNS Resource Channels*

A SMIL presentation allows you to put any number of regions inside a presentation root layout. Find the section labelled Channels in the middle of the Layout view, select *root-layout* to activate the New button and then select New... . This will bring up a dialog box that allows you to define the Channel.

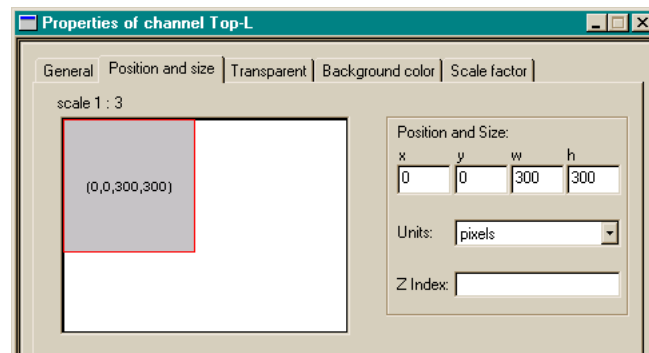


A Channel consists of a name and a type. There may be several Channels of a particular type (such as three audio Channels) in a presentation, but each Channel name must be unique.

In GRiNS/SMIL, the following channel types are supported:

- Null: a channel type without any particular resource policy;
- HTML: a channel holding formatted text;
- Text: channels to render non-streaming text;
- Image: channels to render various types of non-streaming images;
- Sound: audio channels for various audio formats (including RealAudio);
- Video: a channel for various video formats (including RealVideo and Flash).

You should name this Channel *Top-L* (for Top-Left) and give it a type of *Image*. Once you confirm the selection, GRiNS draws a red box containing the new Channel in the default SMIL position: at the top left of the canvas. You can change its size and location in several ways. The first way is to open the property sheet associated with *Top-L* selecting it in the Channels list and then selecting Properties. The Properties dialog contains the following elements:

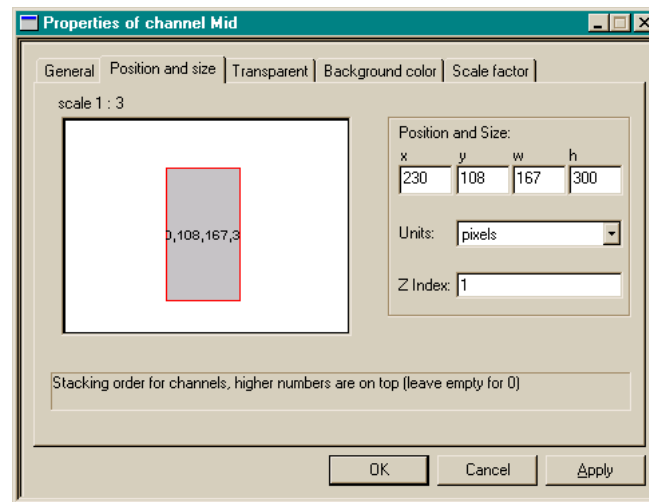


(The placement of the gray window shows the channel location relative to the root window) You can change the location of the window and its size by manipulation of the box or by entering the parameters directly into the Position and Size co-ordinate fields. This is useful if you want pixel-level control over your application.

You should keep *Top-L* positioned at 0,0 pixels (the upper-left part of the canvas), and increase its size to 300x300 pixels. Once you accept the changes, the Top-L Region will be defined and drawn at the appropriate part of the screen.

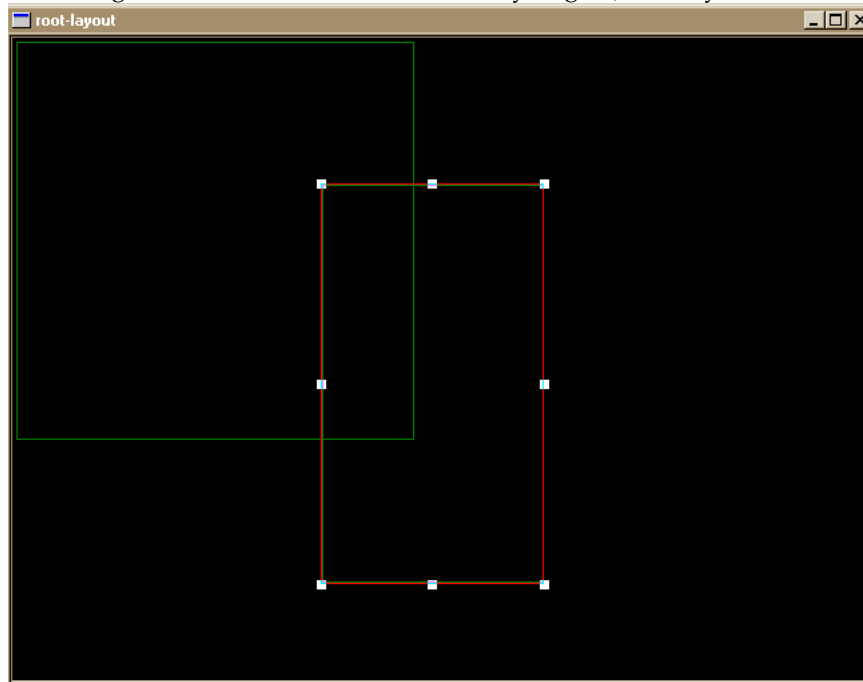
You should now create a second Channel, this time named *Mid* (for Middle). It should also have a type of *Image*. Once the Channel is created, it is drawn with resizing tabs and it is placed in the default position at the top-left of the root window. You can place the mouse anywhere inside the box, and reposition it by keeping the left-mouse key depressed. You can reshape the channel by selecting the resize tabs. You should reposition the channel to (230, 108) and give it dimensions of (167,300).

Go to the *Z Index* selection, and change the Z order from 0 to 1. This means that, in case of an overlap of images, Channels with a higher Z order will be (relatively) on top of the display. You also have the opportunity to change the *scale factor*. There are three choices: if you set the scale to '0', the image will fill the entire window. If you leave it at '1', the image retains its natural size. If you set it to '2', the entire image will be fit inside the window.



Then select either OK or Apply to make the changes.

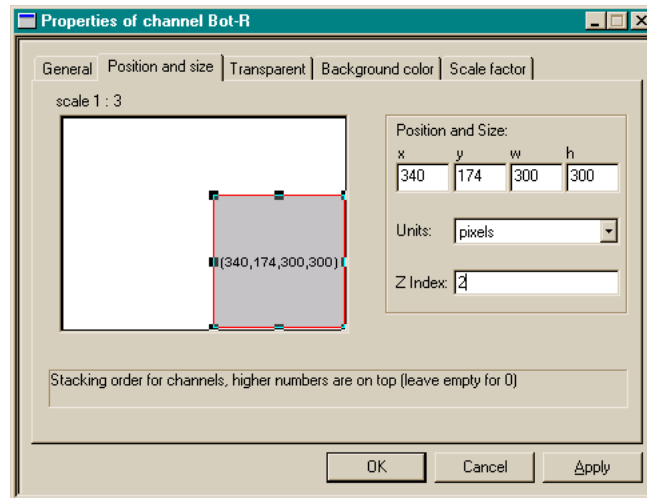
The following illustration shows the view that you get (note: if you don't see the



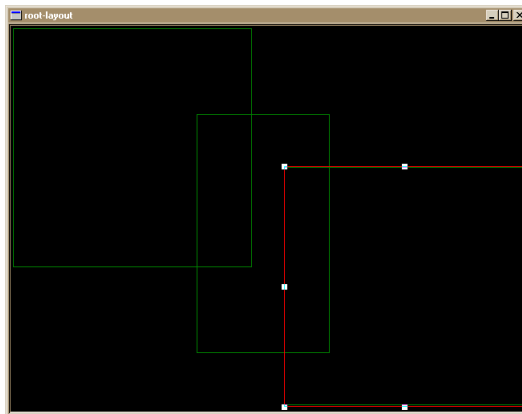
relative outlines, click any of the channel names in the Layout control dialog.

Now, pull up the Properties Dialog for *Mid*, as shown below. (Note: you may need to reselect the Layout view from the Windows menu to re-expose the layout view.)

You can now go on to create the third channel by selecting New from under the Channels list. Give the Channel the name *Bot-R* (bottom right) and give it a type of *Image*. Using either the Properties Dialog approach or the resizing via the mouse, place the Bot-R channel in the bottom right part of the canvas and give it a size of approximately 300x300 pixels. (Note that you may need to expand the GRiNS Editor view to 100% of the screen to see the entire Player window.)

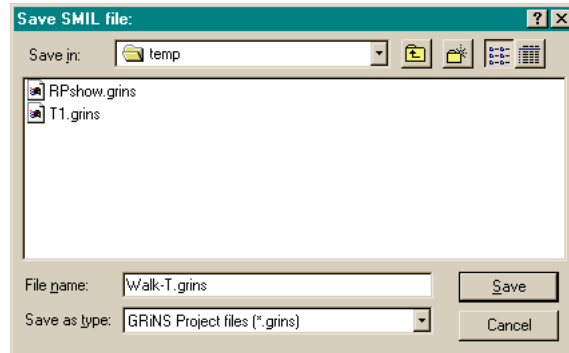


If we used Property-based resizing, we would get the view above.



The presentation now contains one base window and three rendering Channels, named *Top-L*, *Mid*, and *Bot-R*. Each of the Channels is set to render images in the format defined by the author. Note that we don't know if the images will appear in parallel or sequentially, or if each Channel will be used for one or more images. This depends on the specification of the presentation in the Structure and Timeline views.

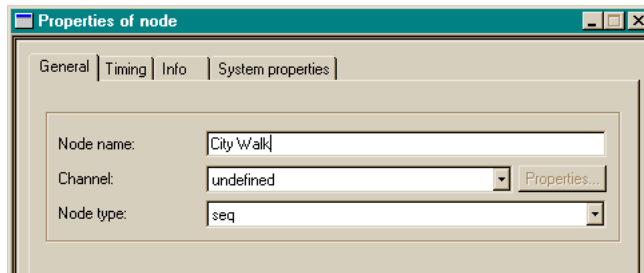
If you haven't done so already, this is a good time to save your work. Use the name *Walk-T.grins*. The settings in the *Save As ...* box are shown on the next page.



### *Defining the Structure and Contents of the Presentation*

The *Walk.grins* presentation used in Tutorial 2 contained a single piece of audio that accompanied a sequence of three images. In SMIL-ese, we need a parallel node containing the audio and a sequential node, with the sequential node providing the images.

We can construct this presentation using the facilities available in the *Structure View*. Close the *Layout View* and expose the *Structure View*. Inside this view, we start by giving the top-level sequential structure a name. We do this in the same way in did in tutorials 1 and 2: by bringing up the structure's *Property* box.



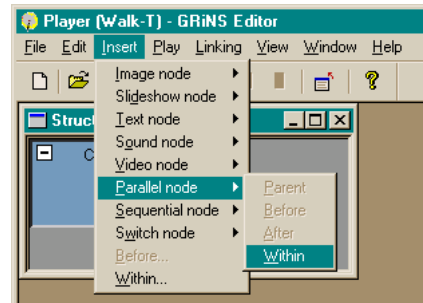
Inside, we label the structure node *City Walk*. The other properties are already filled in by the Editor.

Note:

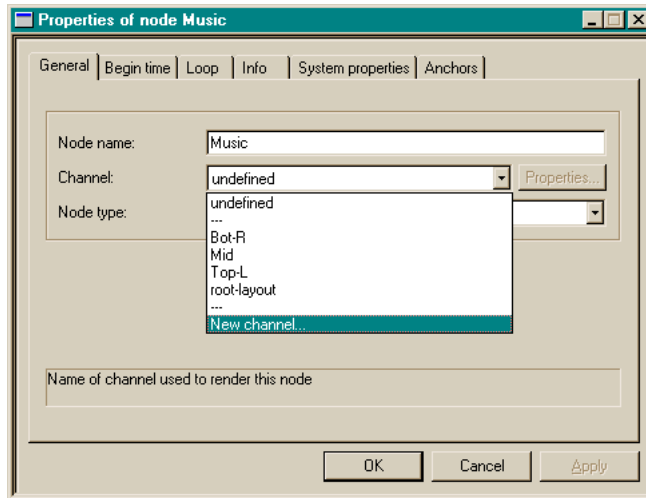
1. The use of a top-level sequential node is optional in the sense that this node could also be transformed directly into a parallel node containing the audio node and the sequential node containing the images. The reason for using the top-level node will become clear in Tutorial 4.

We start the development of the presentation by creating a new node within the *City Walk* container by selecting Insert -> Parallel Node -> Within.

The GRiNS Editor is aware that we only have one option here: you can't put anything before or after the top-level container, so the only choice is to put a new node inside (or within) that container.



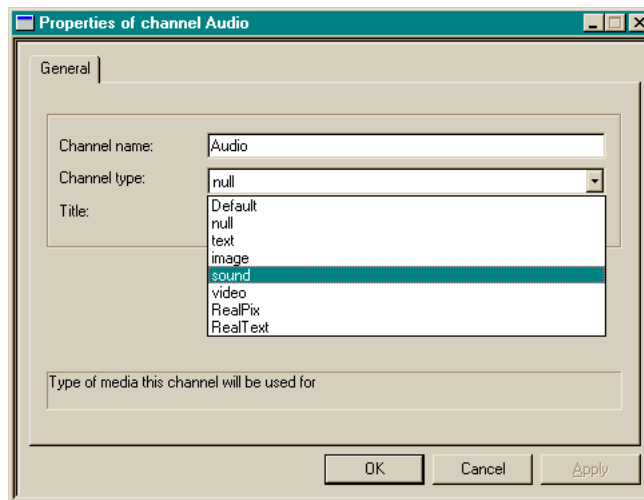
Name the new node *Audio-Visual-Show* and verify that it is a parallel node. Leave the Channel type as undefined.



Continue building the presentation by placing a new node inside the *Audio-Visual-Show*. Select Insert -> Sound Node -> Within, which creates a media object reference instead of a structure node. Name the node *Music*, verify that the type is *ext* (for external) and select *New Channel* under the channel window. The filled-in dialog box is shown.

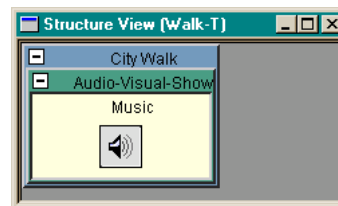


When you select New Channel, you will be given a *Channel name* pop-up box. You should name the new channel *Audio*.



Recall that we haven't defined a layout region for *Audio*, but by creating a new channel, GRiNS has given this channel some default properties. It is a good idea to define the properties that we actually want for the *Audio* channel. We do this by opening the channel's properties, using the Channel properties button on the General tab. Select *sound* as a type, as shown.

After creating the audio node, the Structure view will look as shown at below. What we have defined so far is a presentation structure with a single audio media reference.



We now want to define the sequence of images in parallel with this object.

First, we need to create a sequential structure container. There are several ways of doing this, the most obvious of which is to first select the newly-created Music node and then select Insert -> Sequential Node > After from the right mouse button.

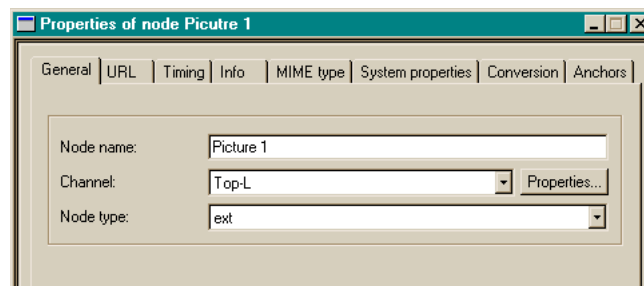
(Since it is not possible to define a node within a media object, this option is not shown.) We will create a sequential structure node after the *Music* object.

Once the new node is created, open that node's Properties Dialog and define the high-level characteristics of the node. Name it *Images*, confirm that it has a type of *seq* (for a sequential node) and leave the channel type undefined.

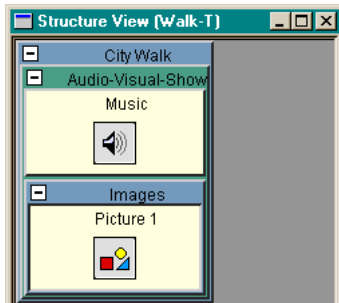
When you accept the changes, the Structure View is:

All that is left to do is to fill the sequential node with the image references. Since we are building a template, we don't want to put new images in the presentation, but elements that will hold images when the template is used.

We start by defining a New node->Within the *Image Sequence* container. Open the Info box associated with this node, and define an image reference. The name is *picture 1* and use *Top-L* as the channel (which we defined earlier in this tutorial).



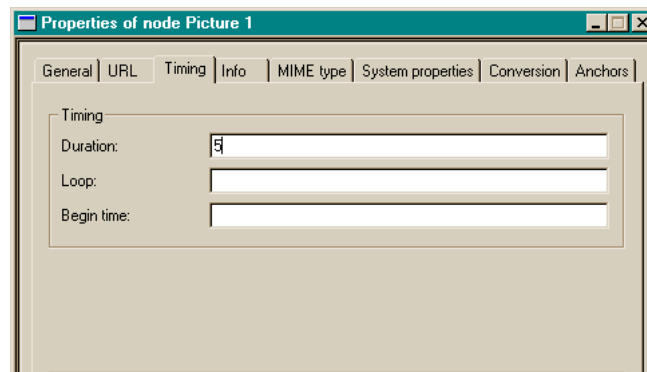
The node is defined as *picture 1*. This is not the name of an actual file (or network URL), but simply a text line that will tell the user of the template what kind of reference should be put in the ultimate presentation. You can also put in the name of a real file (or use a full network URL), but this may create problems for the user during previewing of the application. If you do wish to include a file name or an explicit URL, you can type it in directly or you can browse the local file system for the desired asset.



Once you accept the information in the Property box, a media object reference will be created and the Structure view will be updated with a tan block named *picture 1*, and containing an image icon. (If an actual file name was used, the object reference box would contain a thumbnail image of the referenced file.)

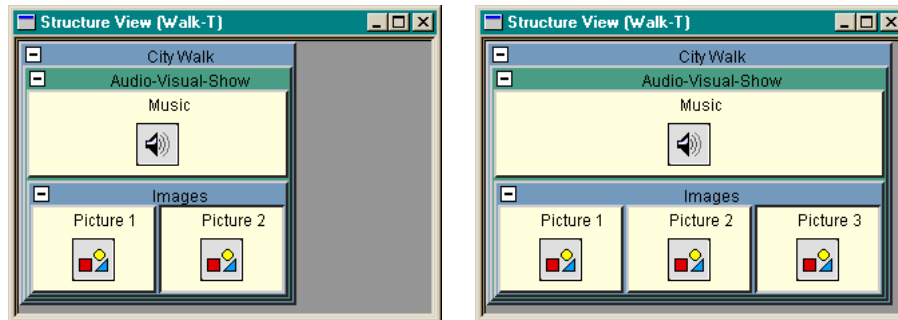
**IMPORTANT:** what you have created is a placeholder object reference for an image. One of the essential features of an image is that it is

timeless — that is, it has no inherent duration. In order to give your template some predictable performance if a user drops an image onto the icon, it is **strongly recommended** that you define a default duration for this placeholder. To do this, open the property sheet for the object (via the Edit menu or via the right mouse button) and set a default duration of, say, five (5) seconds:



Once the *picture 1* image reference is created, we can use the same procedures to create image references for *picture 2* and *picture 3*. You can also use the GRiNS copy and paste facility by selecting copy from under the right mouse (or from the Edit menu) and then Paste After. If you use copy and paste, be sure to open the Property boxes for each node and change the node name and the channel assignment (Mid for *Picture 2* and Bot-R for *Picture 3*).

The resulting Structure Views after each step are:



Make sure that all the information in your template matches that in the examples above. Note that if you used copy/paste to create *picture\_2* and *picture\_3*, the duration of five seconds is inherited by the new objects.

The template is now complete. You should save it for later use.

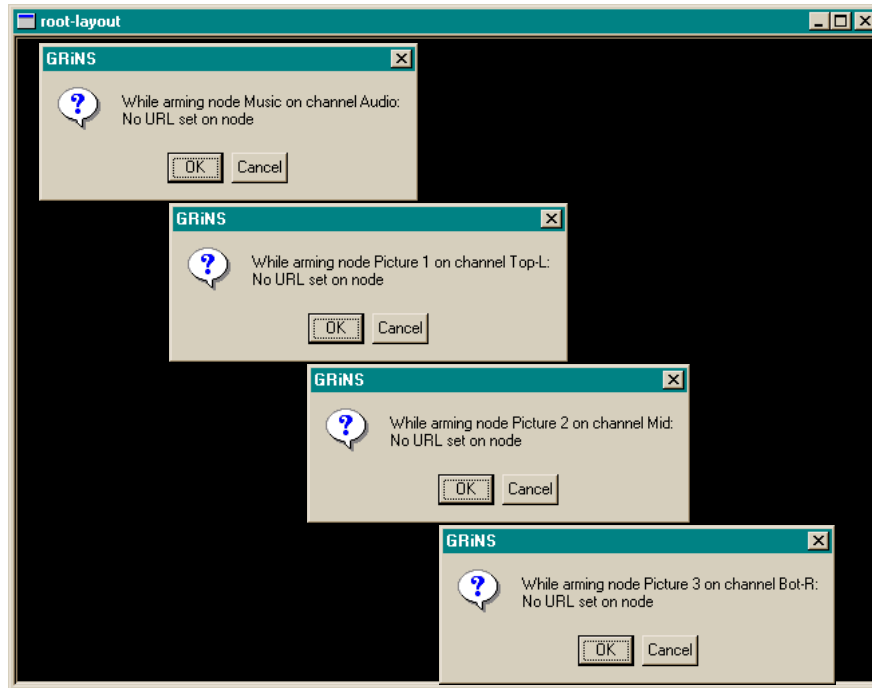
### *Previewing an Empty Presentation*

The *Walk-T.grins* presentation contains all of the structure and object information required to construct the presentation. As it now stands, however, each of the objects only contains a placeholder name. As a result, there isn't much to see in this presentation.

Even though the template is empty, you can still preview the presentation. This may be useful if only a part of the object references have been resolved, or if you are defining a complex presentation where you are not sure what has been defined and what not.

The embedded GRiNS previewer will walk through the presentation-in-progress and inform you of any missing references. For each unresolved reference, a GRiNS pop-up box is displayed with the name of the object that cannot be resolved. If you provided timing information in terms of the duration of objects, then this timing will still be enforced in the presentation.

The image on the following page shows a time-lapsed rendering of what would happen if we previewed our presentation. Note that in an actual preview, you would get the notification boxes one at a time rather than all together, as shown.



### Closing Comments for Tutorial 3

The steps taken to build a simple presentation have been defined in some detail, so that you can get a feeling for the GRiNS general structure. Naturally, we could have made things easier by simply putting all three images on one channel -- or not defining channels at all and having GRiNS define them for us. Hopefully, by taking the somewhat longer road, you have gotten a better idea of how GRiNS works. While this may seem a bit counter-intuitive if you are used to time-line editors, the pay-back comes when you start to edit larger presentations -- or when you need to define alternate content for adaptive presentations.

Take a break and then learn about creating adaptive presentations in Tutorial 5.

## Tutorial 4: Working With Transitions and RealMedia

The facilities available in SMIL-1.0 for placing and activating images provide the basic support needed for many multimedia applications. Some individual data formats extend the facilities of SMIL by providing support for standard image operations such as *fade-in*, *fade-out*, *wipe* and *crossfade*. The most extensive support for this functionality is provided by the RealSystem G2 Player and Real's RealPix format.

In order to use RealPix effectively, several aspects of the format must be understood. One of the most important things to know about RealPix is that it makes use of a timing and layout model that are different from that in SMIL. As a result, care needs to be taken to make sure that the content of a RealPix object is defined in such a way as to allow smooth integration into a SMIL presentation.

If you have a RealSystem G2 player installed on your system, you can render Real's data types in GRiNS documents. You can import any *.smi* file containing RealMedia objects, or you can add RealMedia objects to your presentation. Use the following simple steps:

1. always put RealAudio components in a channel of type sound;
2. put all other RealMedia objects (RealText, RealPix, RealVideo, Flash, etc.), in a channel of type video.

GRiNS/SMIL does NOT contain facilities to create new RealPix files within the editor. If you wish to make extensive use of RealPix (or other RealMedia data items), you should use the GRiNS for RealSystem G2 products, available at:

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





## Tutorial 5: Adapting Contents to Meet the Needs of the User or System Environment

### Overview and Goals

The purpose of this tutorial is to show how GRiNS/SMIL can be used to build *adaptive presentations*. An adaptive presentation is one that can adapt to the needs of the environment when it is being rendered. Adaptation can be based on:

- the speed of the network connection
- the (natural) language preference of the user
- the type of display available to the user
- whether captioning or dubbing of audio material is required

The basic adaptive presentation support available with GRiNS/SMIL matches the facilities available in SMIL. SMIL has a construct known as the *switch* statement, which lets authors define what the alternatives are, and the order in which these alternatives should be evaluated. GRiNS gives the author full control over the SMIL switch.

This tutorial illustrates several types of adaptive control. We start with the control over a single media object and then move to controlling alternative groups of media objects.

You will perform the following steps in this tutorial:

1. Open the Editor and create a document based on the *Extended Slideshow* template;
2. Fill the document with media objects;
3. Edit a system test attribute on one of the media objects to control the object's rendering;
4. Define a Switch container and create a new media reference for an alternative audio object;
5. Define a Switch container to hold alternative GRiNS structure containers.

This tutorial assumes skills developed in *Tutorials 1, 2 and 3*. Please complete those tutorials before starting this one.

## An Introduction to Adaptive Presentations in GRiNS and SMIL

The Web is a heterogeneous place. This heterogeneity is reflected in (at least!) the following aspects:

- *Bandwidth*: the bandwidth available to view a presentation depends on a host of factors that are unknown at the time you create your SMIL document. The user may be connected via a relatively slow modem (14.4Kbps or 28.8Kbps) or via a multi-megabit LAN connection. The server where you place your presentation may be dedicated to your application, or it may be shared with lots of other users. The section of the “information superhighway” between you and your customers may be wide (that is, it can accommodate lots of high-speed connections) or you could live along an Internet side-road. Whatever the potential speeds, your part of the road can be heavily travelled or relatively lightly accessed, all of which impacts the options you have available.
- *Equipment*: there are many different computers and computer configurations available on across the Web. These computers run different operating systems—or many versions of the same operating system—and they have different processing resources available. Some machines will have 800x600 displays, others 1600x1240. Some will have multimedia acceleration hardware, others not. Some machines will have multi-channel audio, others will have one (or none).
- *Users*: It is generally assumed (especially by English speakers) that English is the *lingua franca* of the Internet. For the first generation of use, this was true, but as the Internet becomes more of a world-wide tool, the use of a single language within a presentation can form a serious barrier to broaden the attention of your message. Even within a single language, your personation may be experienced by a wide range of users, including the sight- and hearing-impaired. These last two groups are much wider than you might expect: all of us are sight- and hearing-impaired at one time or another during Web use. If you want to ‘listen’ to the news during a meeting, you are hearing impaired (and would welcome a written summary instead). If you want to ‘watch’ a presentation on your hand-held computer, you may find it more useful to get an audio summary of the show instead.

In each of these cases, a *one size fits all* approach to document design has its limitations. You can significantly enhance the audience for your presentation by providing alternatives to the information content of your work. For example, if you want to show a video to a wide audience, you may want to make both high-resolution and low resolution versions available so that users with high-end systems will get a better experience (without losing customers at the low end). Doing so makes your work more attractive to a wider audience, which will increase your visibility.

The current way of supporting multiple representations of a presentation is to make multiple presentations. This is costly in terms of initial effort, and makes a general presentation very difficult to maintain and enhance.

GRiNS provides an integrated approach to creating these adaptive presentations. You can start with the baseline presentation, and incrementally add enhancements as you go. In so doing, you can make full use of the SMIL *switch* statement.

An important note:

1. GRiNS manages the process of making a presentation with lots of alternative content, but you have to provide the content itself.

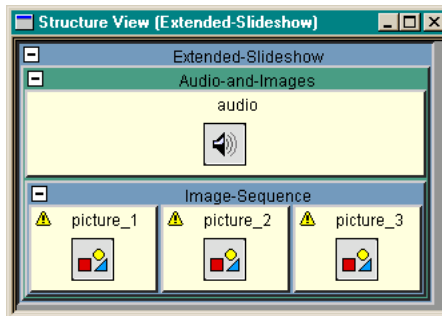
The following sections will show you how to use the GRiNS facilities for creating and maintaining adaptive presentations. This tutorial is set up to teach you a few new GRiNS skills, as well. These include:

- advanced editing techniques
- monitoring bandwidth use during a presentation
- using GRiNS preferences to set presentation environment settings

All of the examples in this presentation build on our running theme of a city walk. In this tutorial, we look at making the walk a more meaningful experience for a wide range of users.

## Creating the Presentation

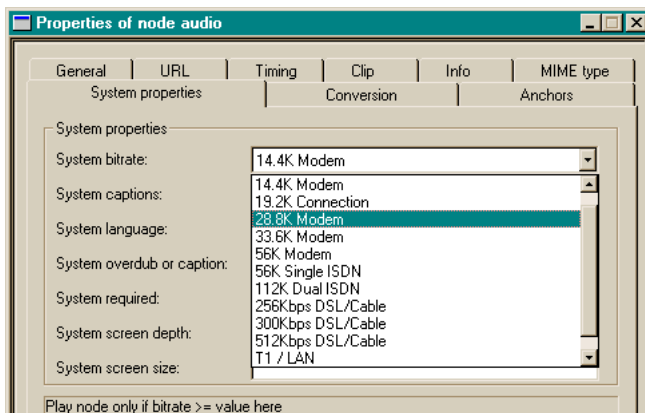
If you have not already done so, start-up the GRiNS Editor; go to the File menu and select New. Select the *Extended-Slideshow* template. The Structure view will look something like:



Now, use your computer's window system to open the directory *Assets*, usually located in the *Tutorials* directory of the GRiNS root folder. Drag the images named *map.jpg*, *h168.jpg* and *h218.jpg* onto *picture\_1*, *picture\_2* and *picture\_3*, and drag the audio object *Atune.aiff* onto the object named *Audio*. Preview the presentation, either as a whole or in part via the Play menu, the shortcut bar or the right mouse.

### Selectively Enabling a Media Object

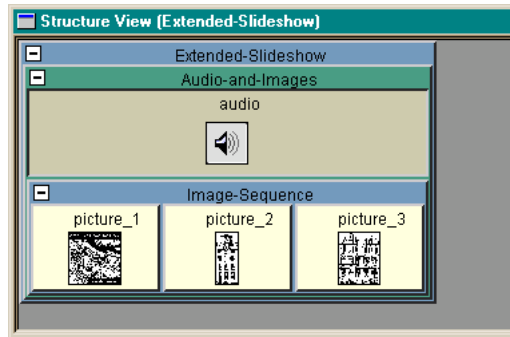
Suppose you wanted to tailor the presentation so that the audio object was rendered only if your user had a network connection rated at 28800 (or greater) speeds. GRiNS provides a simple facility to do this.



Open the Properties dialog associated with the *Audio* object. Locate the item within the System Properties tab that is labelled System bitrate, and set the value to 28800:

This tells the SMIL Player to only render this object if the bitrate set at the client's player is 28800 or higher. (Note

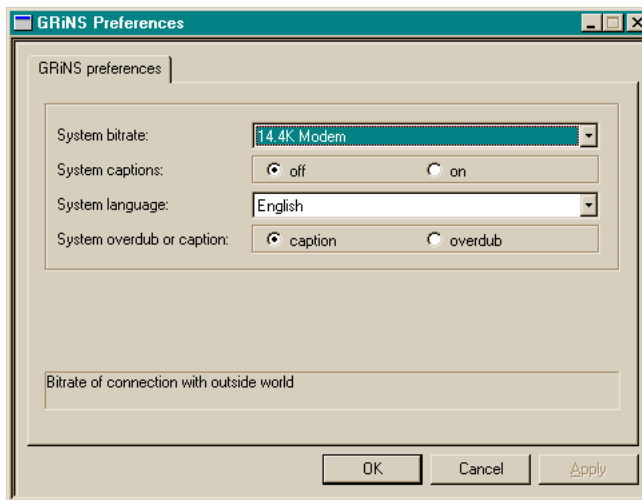
that the actual bitrate will depend on network conditions; this variable refers to a testable setting on the SMIL Player.)



After setting and OK'ing the System bitrate parameter, the structure view may or may not change the color of the Audio component:

In the illustration above, the audio is grayed out because the currently active simulation setting of the GRiNS/SMIL environment has been set to a value lower than that required to play the component.

Now, preview the presentation. If you are running GRiNS as configured at installation, you will see the slideshow but won't hear the audio. Why? Open the GRiNS Preferences dialog by selecting Preferences in the Edit menu. GRiNS will not play the audio file unless your GRiNS preferences have been set to reflect the fact that your connection speed is at least 28k8. (Note: if the GRiNS Preferences had been set to a value greater or equal to 28800, the playback result would be different than that described above.)



Change the *Internet connection speed* to 28800 and play the presentation again. Now you should see and hear all objects.

There are several *System Test Attributes* available to control the SMIL presentation:

- Bitrate: the declared Interconnection bitrate, as defined on the player;
- Captions: only play the object if the user has captions turned on;
- Language: only play the object if the user has specified this as the preferred language;
- Overdub/Captions: display captions instead of dubbed audio, if available;
- Required: only play the object if the named feature is present;
- Screen size: only play the object if the client's screen size is equal to (or larger) than this;
- Screen depth: only play the object if the client's graphics adapter has more bits than this.

The legal values for these attributes are given in the SMIL specification.

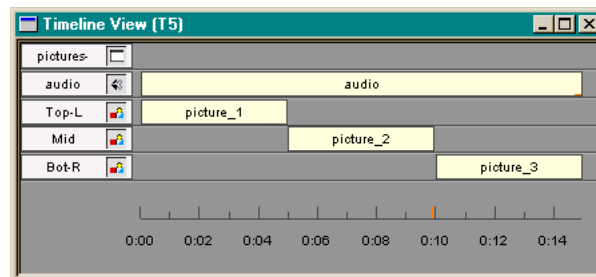
The GRiNS/SMIL allows you to set the connection speed and the following system characteristics via the Preferences dialog:

- *preferred language* (using the list of language names);
- *Closed-Captions preferred over overdub*;
- *Show closed captions*.

### *Viewing Selective Activation in the Timeline View*

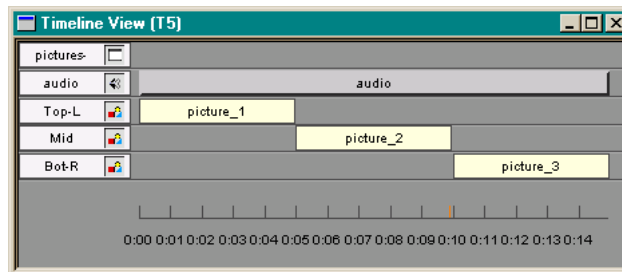
Using Edit -> Preferences, set the *Internet connection speed* to 28800 (or greater).

Now, open the Timeline view. As we saw in *Tutorial 3*, the Timeline view shows a temporal projection of the presentation and is generated automatically by the GRiNS Editor:



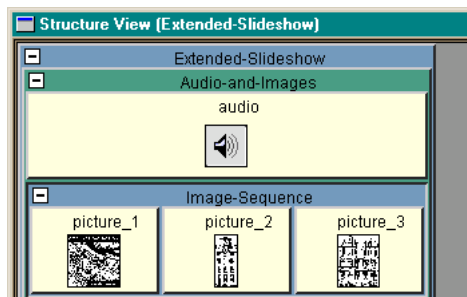
Next, go back to the Preferences dialog and set the *Internet connection speed* to 14400 (or any speed less than 28800). The Timeline view now changes to:

The illustration reflects the fact that the Audio object will not be available at the currently-specified bandwidth setting (it is drawn in grey).



### *Specifying Alternatives for a Group of Media Objects via the Switch*

While SMIL's *test attributes* provide a useful way of controlling object activation, they become more powerful when combined with the SMIL *switch* statement. The Switch allows you to specify a group of alternatives, one of which will be chosen by the SMIL player at runtime.

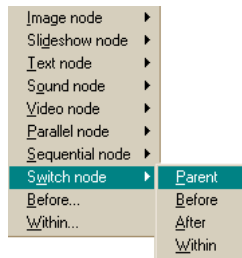


The SMIL Switch is a structure element, much like a sequential or a parallel node. The Switch does not have any rendering behavior in and of itself: it simply tells the Player that a number of candidates must be evaluated at runtime, and to choose the first item which meets all of the selection criteria attached to the objects in the Switch.

To see how the Switch can work within an application, open the Structure view and select the object labelled *Image sequence*, as shown above.

Open the Properties dialog for the node *Image sequence*, and make the following changes:

- change the Name from *Images sequences* to *Images-LQ* (for Low Quality);
- set System Bitrate to Not Set (this is the default and allows playing at any bitrate).



After you accept these changes, select:

Insert -> Switch -> Parent.

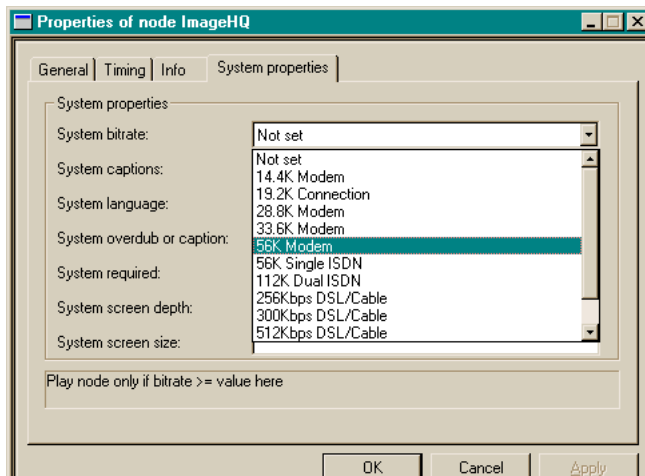
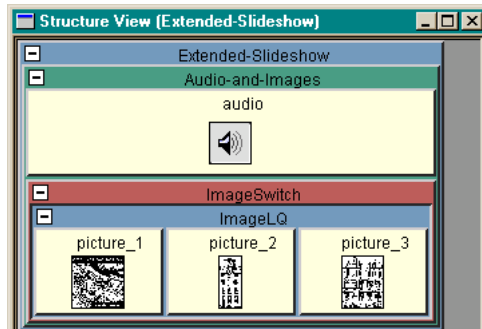
This creates a new parent node to *Image LQ* of type Switch. Name the node *Image Switch*, click OK.

The new Structure view will contain a red Switch node containing the contents of the *Image LQ*, as shown in the View below.

Now, copy the entire object named *Image-LQ*, and paste it before the *Image-LQ* object using the Paste special -> before menu option.

This makes a copy of the entire *Image-LQ* node and places it before the old copy. Go into the Properties dialog for the new version of *Image-LQ* and:

- change the name to *Image-HQ* (for High Quality) in the General tab, and
- set the System bitrate to 56K Modem in the System Properties tab.

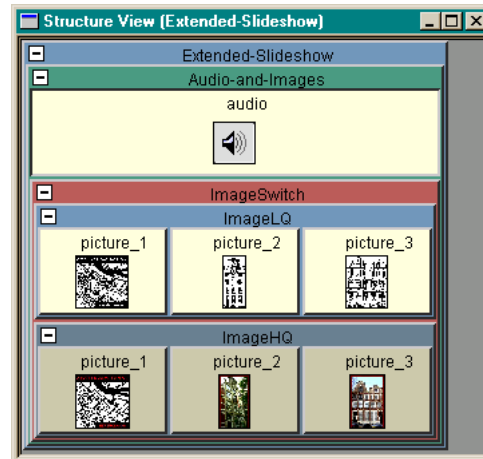




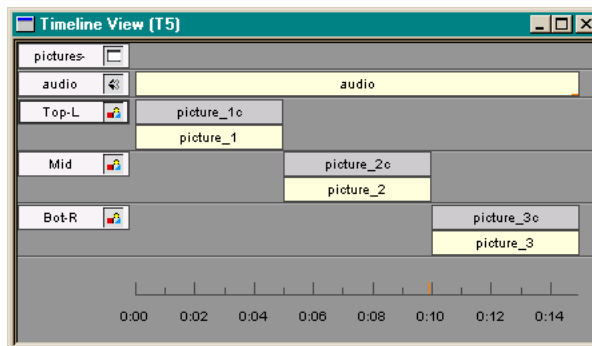
After creating the new structure, you should fill the media object references with new contents. Using GRiNS drag-and-drop, fill the contents of *Images-HQ* with the following objects:

- Picture\_1: replace *map.jpg* with *mapc.jpg* (this is a color version)
- Picture\_2: replace *h168.jpg* with *h168c.jpg*;
- Picture\_3: replace *h218.jpg* with *h218c.jpg*.

The resulting Structure view is shown at right.



You should also give each of the structure containers a new name; this gives each object a unique title, which will become useful when you make detailed timing references or hyperlinks within or across documents. Add a 'c' after each name to create the unique IDs.



To get an idea of what will be happening over time, open the Timeline view. This shows the objects defined per Switch construct. (The bitrate was set to 28800 when this screen shot was made.)

You can now preview the presentation using the various Play options with GRiNS. Depending on the

Preferences settings, you will get three different presentations:

1. If the *Internet connection speed* is set to less than 28800, you will see a sequence of black and white images, but you won't hear anything;

2. If the *speed* is set to between 28800 and 55999, you will see a sequence of black and white images, and hear the audio;
3. If the *speed* is set to 56K or higher, you will hear the audio and see a set of color images.

NOTES:

1. Not renaming media containers may cause your presentation to be rejected by some XML parsers. When copying and pasting collections of node, make sure you give newly-created nodes unique IDs.

### **Implications for Timing, Synchronization and Resource Use**

You can make the contents of a Switch arbitrarily complex. The Switch can also contain nested Switches. When using the Switch, keep the following in mind:

- the Switch will play the first peer-level object that meets all of the enabling conditions set for that node.
- always order the contents of a Switch in a most-resource-intensive to least-resource-intensive order. This will allow your presentation to render correctly.

If you make use of complex levels of Switching, you may need to carefully design your other media objects to make sure that overall timing of your presentation is maintained. In general, avoid the use of explicit durations in non-continuous objects that are played in parallel with Switched elements — GRiNS will adjust timing automatically for you in these cases.

### **Closing Comments for Tutorial 5**

This tutorial was meant to help you understand the basics of creating adaptive presentations. We used controlling bandwidth via connection speed as a convenient example, but there are many other ways to generate adaptive presentations. You should experiment with Language, Captions and alternative media encodings -- such as building a Switch that allows you to substitute a piece of text for a video under low-bandwidth conditions.

## Tutorial 6: Working With Anchors, Links and Hypermedia

### Overview and Goals

The purpose of this tutorial is to show how GRiNS/SMIL can be used to build *hypermedia presentations*. A hypermedia presentation is one that contains both temporal and a-temporal links (that is, links that are active based on time constraints in the presentation, and those that are always active when the object with which they are associated are in view). There are three important concepts that we will treat in this tutorial:

- creating and placing *anchors*
- creating *links* in the presentation to other objects in the same presentation, and
- creating *links* to objects outside the presentation.

The good news about links is that nearly everyone who works with the Web understands the basic concepts. The bad news about HTML links is that they support only a part of the functionality that you need in the time-based presentations that you can make with SMIL. For this reason, we'll spend a page or two on working through the linking concepts supported by GRiNS/SMIL.

You will perform the following steps in this tutorial:

1. Open the Editor and open an existing document containing a collection of structure nodes and media objects;
2. create various types of anchors on these objects;
3. create various types of links between the anchors; and
4. create links to objects outside of the presentation.

This tutorial assumes skills developed in *Tutorials 1* through *5*. Please complete those tutorials before starting this one.

### An Overview of Linking Concepts in Hypermedia

The popularity of HTML has been based, in very large measure, not on the facilities it contains for creating or viewing presentations, but because of its ability to contain links (or, as they are often more popularly know, *hyperlinks*) within and

between documents. It was the link, rather than HTML, which created the semantic fibers that make up the World-Wide Web.

### *HTML Anchors and Links*

One of the things that made HTML links a huge success was their simplicity. Here are eleven lines of HTML that show the basic concepts of HTML links:

```
1  <HTML>
2      <Body>
3          <P>Here is an <A "#name"> anchor </A>.</P>
4          <P>Here is another
5              <A "http://www.oratrix.com/GRiNS">anchor</A>
6              which points to an external document.</P>
7          <P>Here is yet another type of
8              <A "http://www.oratrix.com/GRiNS#buy">anchor</A>.
9          <P id="name">The end.</P>
10     </Body>
11 </HTML>
```

An anchor is a place that is associated with a link. In the examples shown, most of the anchors are denoted by the text between <A> ... </A>. These are what GRiNS/SMIL calls *source anchors*. Most browsers highlight the text between the source anchors, so that the user knows that a link exists here. If you click on the text “anchor” between the <A> ... </A> on line 3, your browser will scroll the presentation to a place with the *id*=“name” string in it (in our case, on line 9). This is also an anchor: in GRiNS/SMIL terms, a *destination anchor*.

As is shown on lines 5 and 8, the destination can be either an entire document, or a point within a document with a destination anchor identified. When someone initiates a link for the anchor in line 8, the browser typically throws away the current presentation and opens the named page, and then scrolls to the place where the *id*=“buy” string is defined.

### *SMIL and GRiNS/SMIL Anchors and Links*

Placing anchors in text is pretty easy in HTML. Placing anchors inside of an image is harder, and placing anchors inside video is very difficult.

In presentations made for a SMIL Player, things are further complicated by the fact that *time* is a factor that you always need to consider. For example, if you have two objects playing in parallel, and one has an anchor with a destination that is at the end of the presentation, what happens to the second object that was playing: does it terminate immediately or does it keep playing?

### GRiNS/SMIL Whole Node and Partial Node Anchors

In order to solve problems like this, and to give links and anchors more meaning in presentations that are time-driven (that is, temporal), SMIL identified a number of linking and anchoring constructs that extended the possibilities of HTML.

GRiNS/SMIL provides extensive support for SMIL hypermedia. In GRiNS/SMIL, an anchor can be attached to the following elements:

1. an entire discrete media object, such as an image or text label;
2. an entire continuous media object, such as a audio or video object;
3. an entire parallel or sequential structure node, containing one or more other nodes;
4. a portion of a discrete media object, such as part of an image or a few words of text;
5. a portion of an audio object (such as between seconds three to five of a ten second audio clip);
6. a portion of a video object, both in terms of time (seconds four to six) and space (the upper left hand corner of the visual space).

Anchors of types 1, 2 and 3 are called *whole node anchors*, while anchors of types 4, 5 and 6 are called *partial node anchors*.

### *Using Anchors and Links in Presentations*

The following sections will show you how to use the GRiNS facilities for creating and maintaining anchors and links. Where all of the examples in this *Tutorial Guide* build on a running theme of a city walk, this tutorial gives you a bit of spending cash to use on that walk. In this tutorial, rather than building an new presentation from scratch or from a template, we open an existing presentation and add in linking functionality.

Note:

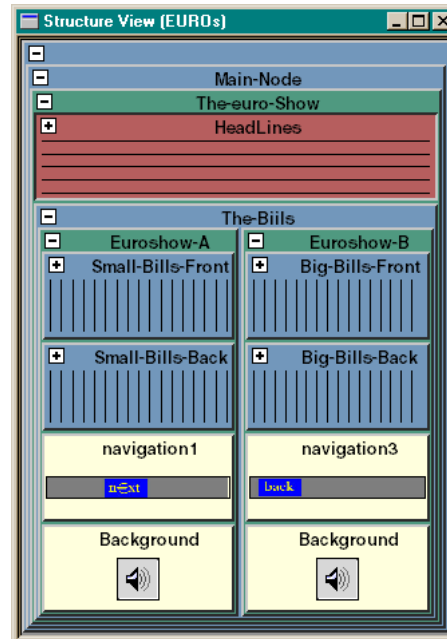
1. Keep the performance implications of links in mind when you design presentations for a streaming environment. If you jump to a destination anchor containing a large temporal media object, several seconds of buffering/pre-loading time may be required before your user sees the object. Since GRiNS/SMIL can't predict when this will happen, a lack of smoothness in the presentation may result.

## Opening the Presentation

If you have not already done so, start-up the GRiNS Editor; either select the Open existing presentation option or go to the File menu and select Open. Select the *EUROs.grins* presentation. The Structure view will look something like:

This presentation has a moderately complex structure. It contains a title that is either rendered as an image or as a text string (the *HeadLines* switch) which is shown in parallel with two objects, each of which contain:

- images of bank notes (both front and back images);
- images that will be used for navigation within the presentation; and
- some background music.



Preview the entire presentation. You will see a series of images showing the bank notes for the common European currency called the *euro*. (Sorry, these images are not suitable for printing!)

## Planning the Links and Anchors

This presentation has been designed to be executed in two parts: first we show the small bills, and then the large ones. Often, it would be nice to let the user jump between these presentations based on a navigation button.

There are several ways to support such navigation in GRiNS/SMIL. You can create links between complete source nodes and a destination node, and you can create links from a part of a node to other nodes in the presentation.

Navigational links are often useful within a presentation, but they can also be used across presentations. We will show this later in this tutorial.

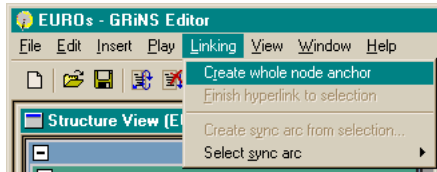
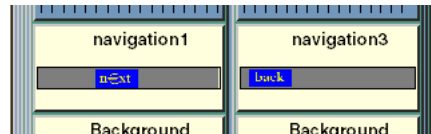
## Creating Simple Links in a Presentation

The most simple form of hyperlink is created by:

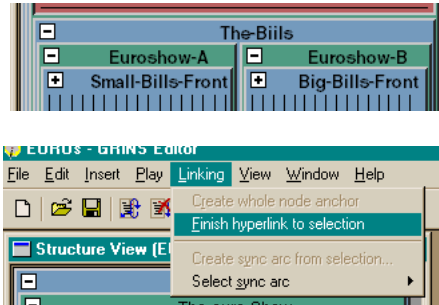
- selecting a node that is to serve as the source anchor for a link,
- creating a *whole node anchor* from the Linking menu,
- selecting the target node for the anchor, and
- creating the link by selecting *Finish hyperlink to selection* from the Linking menu.

You can do this in the open document as follows:

1. select the node labelled *navigation 1* in the *Euroshow A* parallel container;
2. select *Create whole node anchor* from the Linking menu, which tells GRiNS that this will be the source anchor of a link:



3. select the node *Euroshow-B*, which is the node that will be the target of the link we are creating;
4. select Finish hyperlink to selection from the Linking menu. This will complete creation of the link.



Once the link is completed, two new icons are displayed in the structure view. On the navigation 1 node, we see the *source link icon*, which says that a link emanates from this node. On the target, we see a *destination link icon*, which says that a link points to this node.

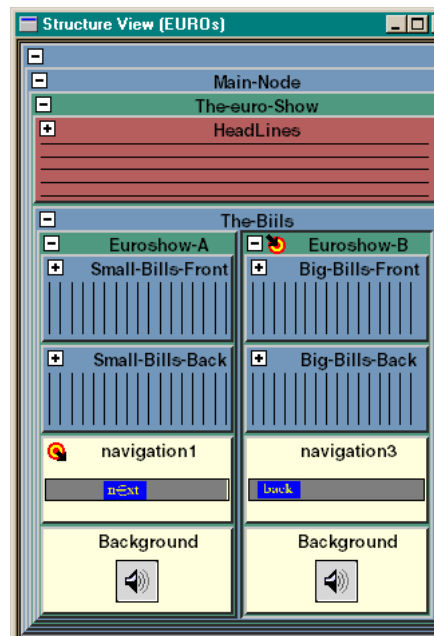
Note that these icons do not show the various source and destination anchors in a presentation, only nodes that are the actual sources and destinations of links.

If you preview the presentation now, you can click on the button labelled *next* and jump to the second fragment of the presentation.

### *Limitations of Simple Links*

The process of creating simple links is only possible under several conditions:

- the *source nodes* of a simple link must be non-temporal media objects (images, text);
- the *destination nodes* of the simple link may be any node in the presentation, but only nodes in the currently open presentation may be selected;





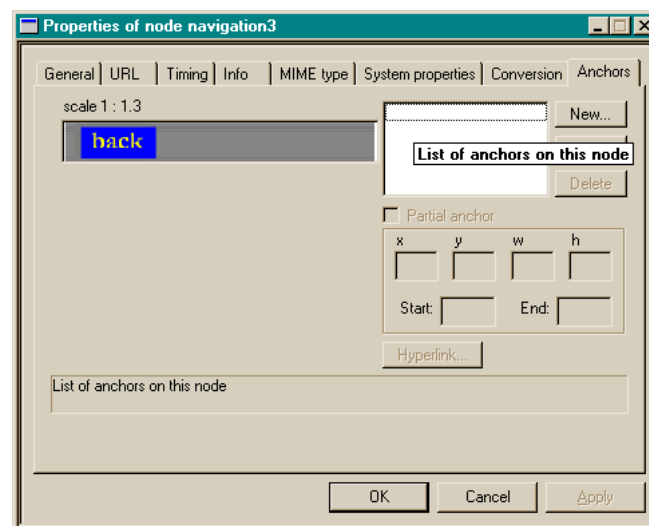
- the links created with simple linking are *uni-directional*, and have SMIL's *replace* semantics. (They are essentially similar to HTML-style links.)

If you can live with these restrictions, you can make presentations with internal navigation quickly and easily.

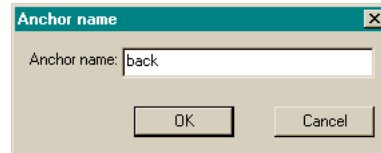
## Extended Linking Facilities

The GRiNS/SMIL editor provides extended linking functions that allows you to create anchors that make only part of a node sensitive to links. We'll show how this works by creating a link from the *navigation 3* node to the *Euroshow-A* node in our euro presentation.

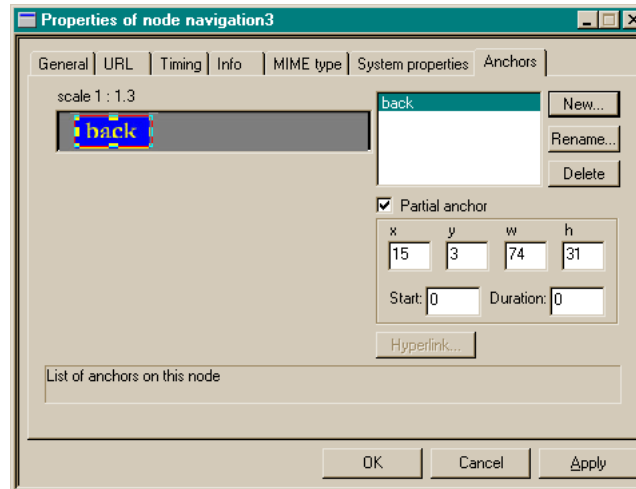
Start by selecting the node *navigation 3* and opening its Properties dialog. Select the Anchors tab. This tab exposes the GRiNS anchor editor.



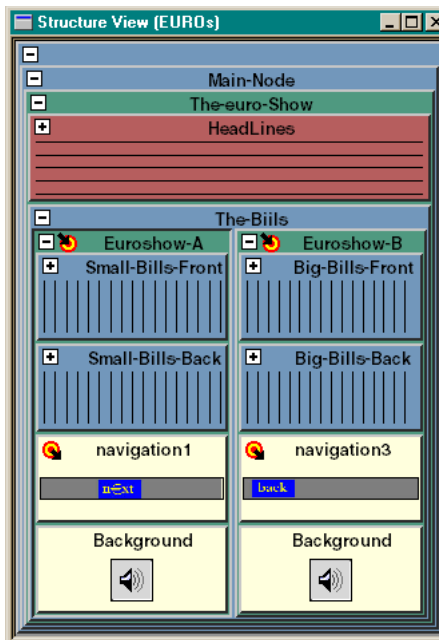
To create a new partial node anchor, select **New** and create an anchor named *back*. You have now created a *whole node source anchor* in this node.



Next, place your mouse over the upper-left corner of the blue button labelled *back* in the image preview area. This results in the creation of a partial node source anchor around the button labelled *back*. You can fine-tune the placement of the anchor using the numeric values (be sure to select **Apply** when you are done). If the source object is a temporal container (such as an audio file or a video) you can also define the points in time that an anchor is active. The values for the source anchor are shown below:



You can finish the creation of the back link by going to the **Structure** view and selecting the target node for the link. Then you specify the **Finish** hyperlink to selection option under the **Linking** menu.



If you had selected the *Euroshow A* node as the destination, the final structure view would be as shown at left.

### Closing Comments for Tutorial 6

This tutorial was meant to help you understand the basics of creating anchors and links within and across presentations. Linking is a powerful feature in creating presentations. SMIL-1.0 provides the basics necessary to get started adding more user control to the messages you place on the Web.



## GRiNS/SMIL Quick Reference Information

### SMIL Compliance Information

The GRiNS Editor for SMIL, Version 1.5 supports the entire SMIL v1.0 specification, with the exception of the constructs listed below. Documents that make use of these constructs are parsed correctly, but the features are ignored during rendering.

- begin and end attributes in the anchor element.
- fit="fill" and fit="scroll" attribute values in region element.
- name="pics-label" attribute value in meta element.
- alt attribute in media object elements.
- fill attribute.

These features are expected to be supported in a future release.

### Supported Media Table

The following chart gives a listing of the media types supported by various versions of GRiNS Editor for SMIL, Version 1.5:

MIME type	Extensions	Windows 95/98/NT	Mac	Linux(4)
audio/basic	au	yes	yes	yes
audio/x-aiff	aiff, aifc, aif	yes	yes	yes
audio/x-wav	wav	yes	yes (1)	yes (1)
image/jpeg	jpeg, jpg	yes	yes	yes
image/png	png	yes (2)	yes	yes
image/tiff	tiff, tif	yes	yes	yes
image/x-portable-anymap	pnm	no	yes	yes

MIME type	Extensions	Windows 95/98/NT	Mac	Linux(4)
image/x-portable-bitmap	pbm	no	yes	yes
image/x-portable-graymap	pgm	no	yes	yes
image/x-portable-pixmap	ppm	no	yes	yes
image/x-rgb	rgb	yes	yes	yes
image/x-xbitmap	xbm	no	yes	yes
	bmp	yes	yes	yes
	ras	yes	no	no
	tga	yes	no	no
video/mpeg	mpeg, mpg	yes	yes	yes
video/quicktime	qt	yes	yes	yes
video/x-msvideo	avi	yes	yes(3)	yes (3)
video/x-sgi-movie	mov	no	no	yes
text/html	html, htm	yes	yes	yes
text/plain	txt	yes	yes	yes

#### Notes

1. Uncompressed WAV only.
2. Support seems to be somewhat buggy.
3. Not all encodings supported.
4. Linux information provided for planning purposed only.

The following chart describes the levels of support provided in the GRiNS/SMIL version for the listed RealMedia data types used in the RealNetworks G2 Player:

RealMedia	Extension	Importable
RealAudio	ra, rm	yes
RealVideo	rv, rm	yes
RealText	rt	yes
RealPix	rp	yes

### References and Links

Please see the Links section of the GRiNS/SMIL Web site ([www.oratrix.com/GRiNS](http://www.oratrix.com/GRiNS)).





## GRiNS Edition Comparison Information

### Packaging Information

The GRiNS Editor is available in several editions, each tailored to the needs of particular groups of authors. These editions are supported under three generic operating systems. This section provides a feature differentiation and compatibility matrix for each of the available editions.

This section compares the various *features* of GRiNS. Please see the next section for a comparison of media types and compatibility with external systems.:

GRiNS Feature	GRiNS /SMIL	GRiNS /QT	GRiNS /G2Pro	GRiNS /G2Lite
Visual Editing Interface	√	√	√	√
Drag & drop editing	√	√	√	√
Integrated Previewer	√	√	√	√
Start Anywhere, Preview Anything	√	√	√	√
Bandwidth-constrained placement	√	√	√	√
Insert / preview all media in native format	√	√	√	√
Insert / preview pre-converted RealMedia	√*	√	√	√
Basic adaptive presentation support	√	√	√	√
Extended adaptive presentation support	√		√	
Basic hyperlink and anchor creation	√	√	√	√
Extended hyperlink and anchor creation	√		√	
Timed-anchor creation	√		√	
Structure view editing	√	√	√	√
Timeline view editing	√		√	
Basic layout editing as attribute	√	√	√	√

<b>GRiNS Feature</b>	<b>GRiNS /SMIL</b>	<b>GRiNS /QT</b>	<b>GRiNS /G2Pro</b>	<b>GRiNS /G2Lite</b>
Extended layout editing via Layout view	√		√	
Sub-region layout support	√	√	√	√
Edit from templates	√	√	√	√
Create new templates	√		√	
Basic G2 attribute suport	√*	√	√	√
Extended G2 attribute support			√	
Basic bandwidth analysis and preview	√	√	√	√
Extended bandwidth analysis and preview			√	
Basic conversion of native datatypes to RealMedia			√	√
Extended automatic conversion to RealMedia			√	
Open/edit any existing SMIL presentation	√*		√	√
Automatically generate Web Page for presentation			√	√
Automatically generate external G2 presentation			√	√
Automatically generate G2 presentation within an HTML Page			√	
RealServer Upload Support			√	√

**NOTE:**

1. Items tagged with a '\*' require that the RealNetworks G2 player be installed.

The available versions of the GRiNS/SMIL 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/>