



GRiNS Tutorials Guide

**GRiNS for SMIL
Editor/Player Version 1.0
Windows, Macintosh and UNIX**



© 1999 Oratrix Development bv. All rights reserved.

GRiNS for SMIL v1.0 Tutorial Guide for Windows, Macintosh and UNIX. August, 1999.

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 for SMIL*, *GRiNS/SMIL*, *GRiNS Editor* and *GRiNS Player* 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 first version of the *GRiNS Tutorials Guide* for Version 1.0 of GRiNS for SMIL. All of the information has been verified, but incremental product updates may impact part of this guide.

This version of the *GRiNS for SMIL 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 for SMIL 1.0-win32-0 version (for Windows-95|98|NT-4). While the look of other versions of GRiNS are slightly different because of adherence to common conventions on those other environments, the functionality described is similar for all versions of GRiNS. In order to reduce document size, only images from the Windows version have been included in this document.

We welcome your questions on GRiNS for SMIL and comments on this documentation. Please submit all questions and comments to our support desk at grins-support@oratrix.com. We maintain a list server dedicated to sharing experiences among GRiNS/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.

Table of Contents

GRiNS Tutorials Roadmap	vii
Distribution Package Contents	vii
GRiNS Tutorials Guide	viii
Introduction	1
The GRiNS Editor for SMIL	1
A Note on the Illustrations Used in the Guide	3
<i>Apple Macintosh</i>	3
<i>SGI and SUN UNIX notes</i>	4
Tutorial 1: Creating a Simple Presentation	5
Overview and Goals	5
Open the GRiNS Editor and select the Basic-Slideshow.SMIL template	6
Creating the Presentation	8
Editing in the GRiNS Structure View	9
<i>Understanding the Structure View</i>	10
<i>Fill in the First Image for Use in the Show</i>	11
<i>Fill in the Remaining Images for Use in the Show</i>	13
Playing the Presentation	15
Further Editing Options	15
<i>Changing Detailed Properties of a Node</i>	16
<i>Reordering the Presentation via Cut and Paste</i>	16
<i>Expanding the Presentation by Using Copy/Paste</i>	18
Closing Comments on Tutorial 1	19
Tutorial 2: Extending an Existing Presentation	21
Previewing the Results	21
Open the Walk.smil presentation	22
Understanding the GRiNS Structure View	23
<i>Adding a new media object reference</i>	24
Manipulating Structure and Data Objects	26
Previewing the Results	27
Fine-Tuning the Timing of a Presentation	28
<i>Setting an Object Reference to "Loop"</i>	28

<i>Adjusting the loop length</i>	29
<i>Setting an explicit duration</i>	30
<i>Explicitly Selecting the End of a Parallel Group</i>	32
<i>Introduction to the Timeline View</i>	35
<i>Fine-Tuning via Synchronization Arcs (Sync Arcs)</i>	36
<i>Better Understanding Timing in Parallel and Sequential Structure Nodes</i>	39
<i>Selecting Layout Regions</i>	41
Closing Comments on Tutorial 2	42
Tutorial 3: Creating Templates, Working With Regions and Using GRiNS Resource Channels	43
An Introduction to Layout Manipulation in GRiNS and SMIL	43
<i>Layout and the Walk.smil presentation</i>	44
Open the GRiNS Editor and select the EMPTY.SMIL template	46
Define Regions for Displaying Presentation Components	47
<i>Define SMIL Regions via GRiNS Resource Channels</i>	48
Defining the Structure and Contents of the Presentation	53
Previewing the Presentation	59
Closing Comments for Tutorial 3	61
Tutorial 4: Working With Anchors, Links and Hypermedia	63
Tutorial 5: Adapting Contents to Meet the Needs of the User or System Environment	65
GRiNS Quick Reference Information	67
SMIL Compliance Information	67
Supported Media Table	67
Support for RealMedia (G2) Data Types	70
Access methods	70
References and Links	71

GRiNS Tutorials Roadmap

Thank you for downloading the GRiNS for SMIL (GRiNS/SMIL) toolset for playing and (optionally) creating SMIL presentations.

This publication will help you understand how GRiNS works, and how it can help you to create high-quality multimedia presentations for the Web easily and effectively.

Distribution Package Contents

The GRiNS distribution package consists of the following components:

- *GRiNS for 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, some of which are referenced in the *Quick Start Guide*;
- *GRiNS-Icons*: a directory containing Icons used by the GRiNS Editor;
- *Software*: depending on the distribution you downloaded, a GRiNS distribution for Windows-95/98/NT, the Apple Macintosh or UNIX.

Before starting the tutorials in this guide, you must download and install a version of the GRiNS Editor. A license key is required to activate this software.

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

GRiNS Tutorials Guide

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

1. *Building a Simple Presentation*: a basic introduction to the GRiNS Editor, showing you how to take a simple template and fill in data objects.
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. *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.
5. *Hypermedia Support in GRiNS*: Linear presentations are great, until you (and your customers) see too many of them. GRiNS provides support for the full hypermedia features of SMIL, as shown in this tutorial.

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.

After you get experience working with GRiNS and SMIL, you should read the *GRiNS for SMIL User's Guide* and the *GRiNS for SMIL Reference Manual* for more detailed information on how SMIL and GRiNS can support your presentation development needs. (These manuals are available with purchased copies of the Editor only.)

Introduction

The GRiNS Editor for SMIL

This guide will help you to learn how to create SMIL documents using GRiNS.

SMIL is a multimedia description language that was defined (and is maintained and further developed) by W3C: the World-Wide Web Consortium. The GRiNS Player 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 will be delivered to users across the Internet via a streaming multimedia player, such as the RealSystem G2 player. While you don't need streaming media to define a SMIL presentation, placing your SMIL presentation on a streaming server and having your users access it via a streaming player (such as G2) will probably provide them with the most effective presentation over the current Internet.

You can also create local presentations using GRiNS — that is, presentations that do not need to be accessed via a streaming server — that can be played back using any SMIL compatible player (such as the GRiNS for SMIL Player or RealSystem G2). 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.

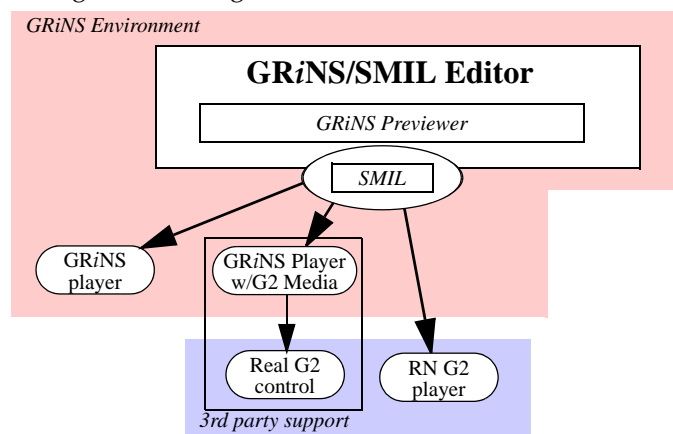
The *GRiNS for SMIL* Editor provides a single interface that allows you to create presentations to any SMIL compliant player. GRiNS uses an embedded previewer that helps you define your presentation quickly and correctly. In addition, you can edit any existing SMIL document (whether initially created with GRiNS or not).

While all SMIL-compliant players understand and process the SMIL-1.0 language, not all SMIL players support the same internal data types. Some SMIL players, such as RealSystem G2, have been optimized for streaming media formats like RealAudio, RealFlash, RealPix, RealText, and RealVideo. Other players, such as

the GRiNS/SMIL Player, allow non-streaming data types (such as HTML, MPEG video and audio, and a host of image formats).

Before you create a document, you should first develop a clear understanding of your target audience. If your audience is spread across the Internet and if you intend to use a streaming server to deliver your presentation, you should restrict the data types you use to those supported by RealNetworks. If you don't need streaming services, then you can also use other data types in your presentation.

The following illustration gives an overview of GRiNS/SMIL V1.0:



The GRiNS Editor for SMIL is a flexible interface for creating, maintaining and previewing SMIL presentations. You can target your presentation to the:

- *RealSystem G2 Player*: a full streaming SMIL-1.0 compliant player provided by RealNetworks (available for Windows and Macintosh);
- *GRiNS/SMIL Player with RealMedia Support*: the standard GRiNS/SMIL Player with extensions to render RealMedia data types using Real's G2 control (available for Windows and Macintosh);
- *GRiNS/SMIL Player*: a full non-streaming SMIL-1.0 compliant player (available for Windows, Macintosh and UNIX).

The choice of target player will depend on your application. To reach the widest possible audience, tailor your presentation to use only RealMedia data types and

the RealSystem G2 player. To be able to integrate HTML into your SMIL presentation, or to make a SMIL presentation that can also be viewed on a UNIX platform, use the GRiNS/SMIL Player. If the G2 player is installed, you can also use all of the RealMedia data types in your presentation — but only on platforms for which RealMedia is supported.

Notes:

1. In order to render Real's proprietary RealMedia data types (RealAudio, RealVideo, RealFlash, RealText and RealPix) — either via the GRiNS Player or via Real's G2 player — a version of G2 must be installed on the user's system. The RealSystem G2 player is not delivered with the GRiNS distribution. It must be obtained separately from Real Networks at: www.real.com.
2. Presentations created with the GRiNS/SMIL Editor can be rendered using any other SMIL-1.0 compliant player, but check the player's documentation to determine the data types it supports.

A Note on the Illustrations Used in the Guide

Although GRiNS is available for Windows, Macintosh and UNIX platforms, this Tutorial Guide provides illustrations and examples drawn only from the Windows version. We have done this to minimize the document's size. 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 UNIX versions should keep the differences summarized below in mind.

Apple Macintosh

The Mac has only a single-button mouse. Where the text refers to a selection using the right mouse button, you should substitute a CTRL-click selection instead. 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.

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: Creating a Simple Presentation

Overview and Goals

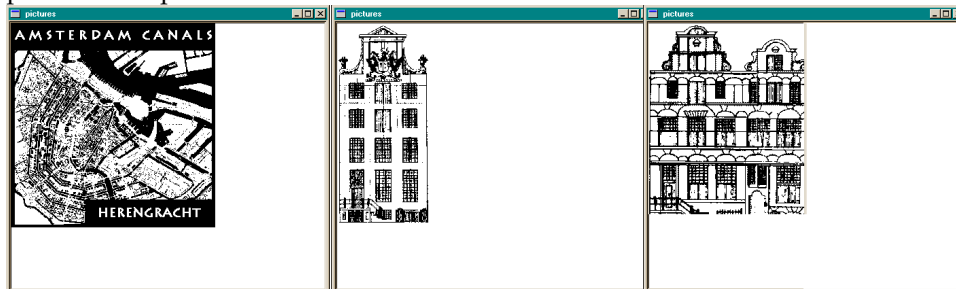
The purpose of this tutorial is to create a very simple SMIL file that consists of a sequence of objects that get rendered onto a default area of the screen. This presentation makes use of a GRiNS Template document.

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

1. Open the editor and select the Basic-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. Play the presentation

We do this step-by-step so that you can follow the process in great detail.

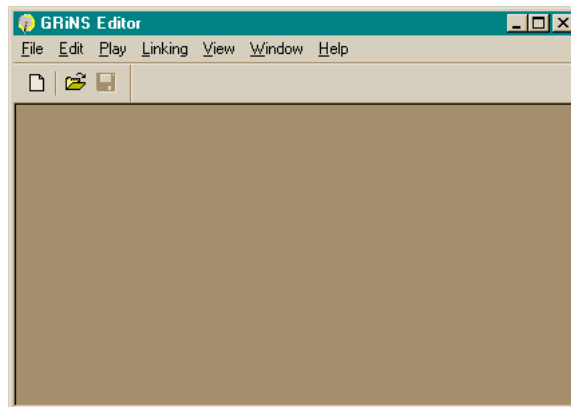
The result of building the first part of this tutorial is a series of three pictures displayed in the (player-dependent) default location on the screen. A thumbnail presentation preview is:



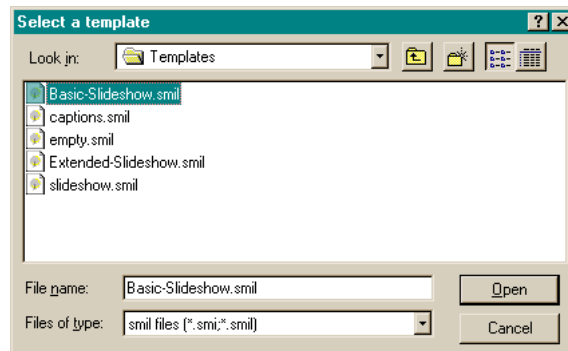
After some modifications, the presentation will show a fourth image, which is a copy of the first.

Open the GRiNS Editor and select the Basic-Slideshow.SMIL template

If you have not already done so, start-up the GRiNS Editor; this will give you the following window:

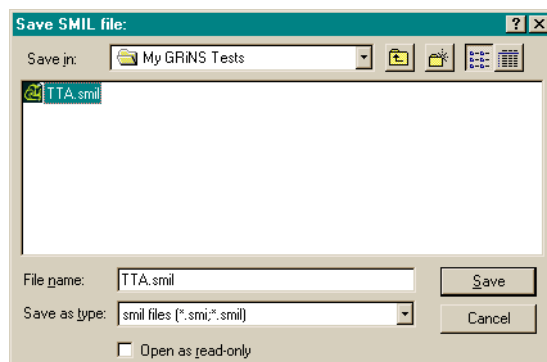
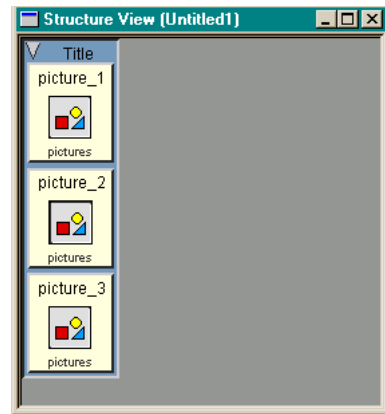


Next, go to the File menu and select New, which allows you to select a GRiNS Template file. In this tutorial, select the *Basic-Slideshow.smil* file, as shown below:



This will open the SMIL document *Basic-Slideshow*, giving you the Editor's basic view, shown at right. In this view, we see you blue outer container labelled *Title* and a series of three light-colored nodes called *picture_1*, *picture_2*, and *picture_3*. At the top-left of the structure container is a small triangle, pointed downward. Give the triangle a click to see one of GRiNS/SMIL's information hiding features.

(For SMIL fans: the blue structure container is a SEQ — or sequential — container.)



Before doing anything else, it is good practice at this point to immediately save you newly-opened file somewhere. We recommend that you create a folder called *My GRiNS Tests* to save your work-in-progress. (You may also put the file elsewhere, of course.) Select Save as ... from the File and name your file *TTA.smil*.

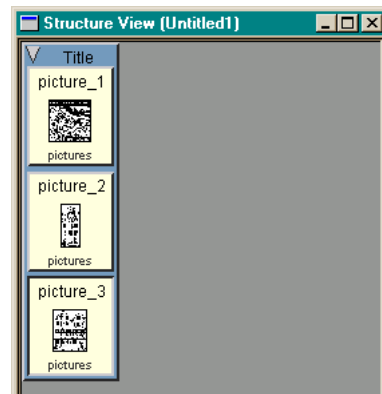
NOTES:

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 of GRiNS materials;
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.

Creating the Presentation

The *Basic-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. Open the Data folder in the GRiNS root directory (typically located at *c:/Program Files/Oratrix/GRiNS 1.0/Data* on Windows platforms);
2. Select the image *map.jpg* from the folder and drag it on to the light-colored box labelled *Picture_1*;
3. Select the image *h168.jpg* and drag it on to the box labelled *Picture_2*;
4. Select the image *h218.jpg* and drag it on to the box labelled *Picture_3*;
5. Play the presentation by selecting Play from the Play menu.



That's it: you created your first GRiNS/SMIL presentation!

GRiNS allows you to perform some special functions that help make editing easier and more productive. Many of them are introduced in this tutorials guide.

To get a quick taste of what GRiNS 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 you presentation with a minimum of authoring effort.

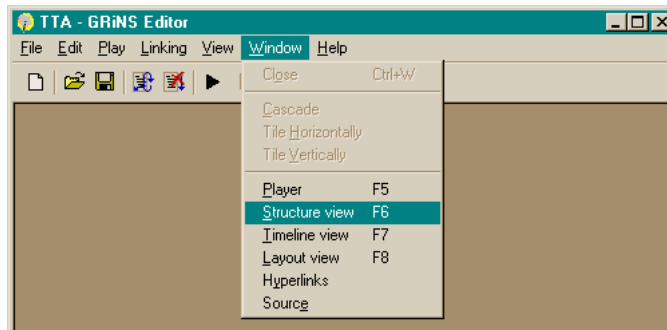
Editing in the GRiNS Structure View

GRiNS provides an intuitive interface to creating presentations. To understand the full power of GRiNS, it is important to know a bit of background on SMIL.

SMIL is a structured presentation language. This means that you define a number of structure containers that are used to hold either data objects or other structure containers. The two types of structure containers available are the sequential container (called a *SEQ* node in SMIL) and the parallel container (called the *PAR* node in SMIL). Elements placed in a *SEQ* container are activated sequentially, while elements placed in a *PAR* container are activated in parallel.

It is important to realize that the activation times of all of the components in a presentation is determined by the nested SEQ and PAR structure of the SMIL definition. To reflect this, the GRiNS Editor uses a special view called the Structure View to represent the structure of a SMIL presentation. As we will show in the first three tutorials, you can specify an entire SMIL presentation using only the Structure View.

The Structure View is the default view in the GRiNS Editor. If it has been closed (for whatever reason) you can always explicitly open this view by going to the Window menu and selecting Structure View, as shown below:

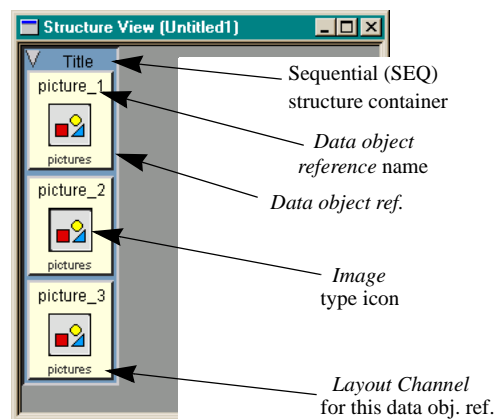


For an introduction to this and other Views within GRiNS, you should consult the *GRiNS Quick Start Guide*, available from the GRiNS Web site:

www.oratrix.com/GRiNS/.

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 Basic-Slideshow.smil template, the following detailed information is shown in the Structure view:



The presentation consists of three data object references (the light-colored boxes) and one structure node (the blue box labelled *Title*). Each of the data 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.

Because this is a template file, the structure of the application is complete, but you still need to fill in the individual image file names and locations. Once you fill an empty data object reference with an actual object, a thumbnail image of that object is placed in the structure view. The name used in the object reference (such as *picture_1* is not the name of the file, but an ID used in the presentation.

Fill in the First Image for Use in the Show

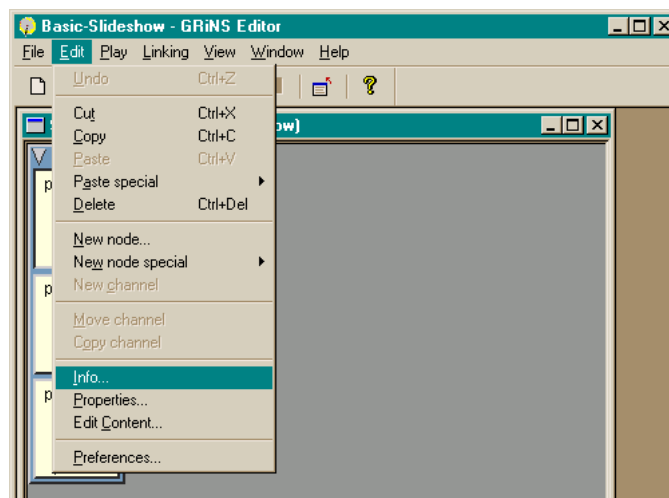
GRiNS gives you two mechanisms to associate data objects with object references. First, you can drag-and-drop a media item on the data object reference node. This works for any type of media object supported by the Editor. A second method for associating media objects with a node is by bringing up a set of property sheets that let you fill in the name (and possibly location) of the data object you would insert.

By opening the object reference (the light-colored box), you gain access to various attributes associated with this reference. The more attributes you define, the more control you have over your presentation.

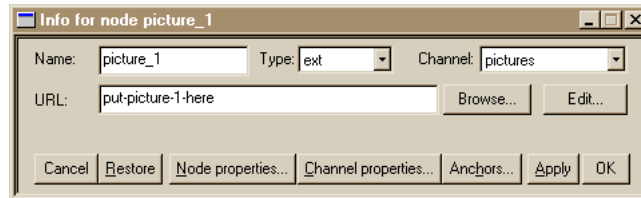
The attributes for any object are partitioned into three sets of property sheets:

- Info sheet: this sheet lets you define the basic properties of an object;
- Properties sheet: this sheet lets you define detailed attributes for an object;
- Channel sheet: this sheet lets you define information on the resources to be used by this object.

To open the Info sheet for any node, go to the Edit menu, and select Info, as shown below:



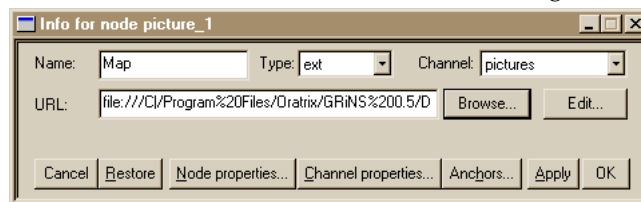
This gives a dialog box where you define the basic properties of the reference:



(You can also bring up an Info sheet by double-clicking a node in the Structure view.)

You should do the following:

1. change the name of the node from *picture_1* to *MAP*. This gives a name to the data object reference, but it does not identify a particular object to be associated with this reference. Some SMIL players use this information to help you build a presentation table of contents.
2. select the Browse button to select a data object to be associated with this reference. Use the file named *map.jpg*, which is located in the *Data* folder provided with these tutorials. The result is the following Info box:

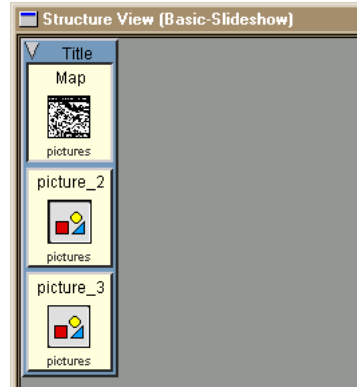


3. confirm all selections

Note that the file name is a fully qualified URL, not a local file system reference. GRiNS uses URLs exclusively. For multi-platform compatibility, it is best to select files using the Browse button rather than filling in the file name directly in the URL field.

Note also that the Info dialog contains an Edit button -- you can use this to bring up the editor defined on your system for the associated data type. (On Windows platforms, this is the program associated with the file extension of the object's name.)

After filling in the details, the Structure View should look as follows:



The name of the top data object reference has changed to *MAP*, a small thumb-nail of the selected image is displayed.

Notes:

1. When naming data objects, always use a URL if you are going to enter the name directly.
2. If you want to select a file on the local system, use the Browse function, and select the file via the embedded file browser; this will ensure that naming is entered correctly.
3. If you type in a local file name directly, you should only do this for files in the same directory as the SMIL file -- do not use '\' or '.' in a file name, as its interpretation may be unpredictable in various internal Windows functions.

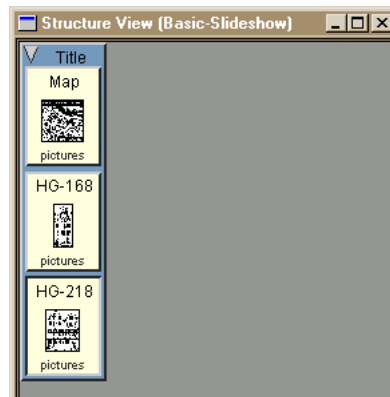
Fill in the Remaining Images for Use in the Show

Using the same sequence as in section above, fill in the data object references for *picture_2* and *picture_3* as follows:

1. select the *picture_2* object once, watch that it is highlighted;
2. using the right mouse button, select Info... (or use the Tools -> Info menu combination).
3. rename the object reference HG-168 and set the file to h168.jpg in the Data folder.
4. confirm these selections and next select *picture_3*.
5. using the right mouse button, again select Info.

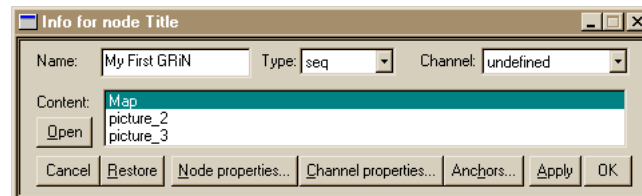
6. rename the object reference HG-218 and set the file to h218.jpg.
7. confirm all selections.

The result is the following Structure View:



Finally, you should give this structure container a meaningful title. (This is optional, but it provides information that some players can use to index your presentation.) To do this, locate the blue box labelled *Title* and double-click it. The box you have just selected is a sequential structure container -- it holds the three objects, and tells the SMIL player to play them in sequence.

Structure nodes have properties just as data object references have. You can also use the Info command to rename this container *My First GRiN*.



In this Info box, we see that there is no pointer to a file or URL, but a list of the objects that are contained inside this structure node.

Playing the Presentation

In the *Basic-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.

NOTES:

1. In our examples, we encourage you to play the entire presentation. The GRiNS Editor also allows selective playing: you can either select one node (the Play node entry in the Play menu) or identify a starting point other than the beginning by selecting Play from here in the Play menu.

Further Editing Options

Filling in the template is a handy way to start building presentations quickly, but the Editor provides you with many other ways to manipulate a presentation.

In this section, we show three simple editing operations that will start to develop your GRiNS skills.

- you can change some of the detailed attributes of an object, such as its duration
- you can reorder elements in a presentation by cutting and pasting
- you can reuse elements in a presentation by copying and pasting

Changing Detailed Properties of a Node

Select the second data object reference (named *HG-168*) by clicking on it once. Using either the right mouse button, or the Edit menu, select Properties, which brings up the following window:

Property	Current Value	Default	Explanation
Node name	HG-168		Name given to a node
Channel	pictures	undefined	Name of channel used to render this node (or its descendants)
URL	file:///C:/Progr...		URL for the data of this node
Duration	2	0	Duration of a node or subtree (in seconds)
Loop		1	Number of times the node should loop (0 is indefinite)
Title			The title of this object (documentary)
Abstract			A brief description of the contents (documentary)
Author			Author (documentary)
Copyright			Copyright of the item (documentary)
Comment			A comment (authoring convenience only)
Screen	Default	undefined	Screen for this node and its descendants. Authoring convenience only, controls ...
MIME type			MIME type of media object
System bitrate		0	Play node only if bitrate >= value here
System captions	Default	off	Play node only if end-user wants captions
System language			Play node only if end-user prefers this language
System overdub or caption			Play node only depending on end user overdub/caption preference
System required			Play node only if player supports these features
System screen size		0 0	Play node only if screen bigger than this
System screen depth		0	Play node only if screen has more bits than this

Duration: Reset

Cancel Restore Apply OK

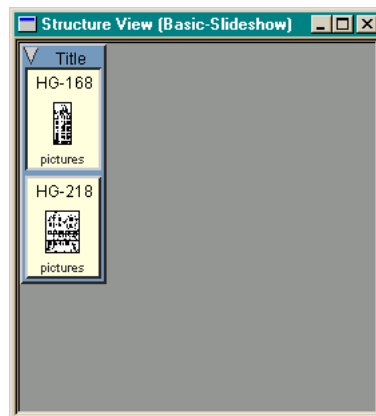
Using either the arrow keys or the mouse, select the line labelled *Duration*. In the attribute editing area at the bottom of the box, replace the value 5 with 2; this reduces the time that this object is visible from five seconds to two seconds. Confirm with OK. Now, run via Play. The first image will appear for five seconds, the second for two seconds and the third for five seconds. The third image is replaced by the white background color at the end of the presentation.

Reordering the Presentation via Cut and Paste

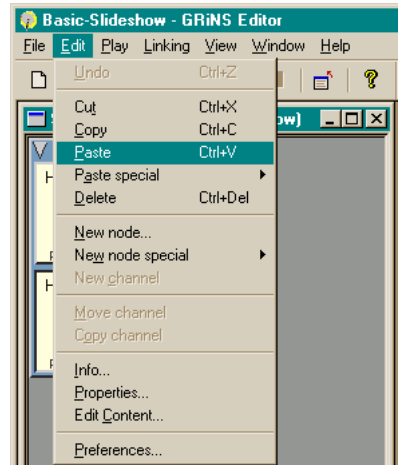
Close the player view (labelled *Pictures*) by using standard Windows techniques. This should fully expose the Structure View again. In this example, we will move the first image (the map of Amsterdam) from the front of the presentation to the end

using the GRiNS cut-and-paste functionality. Select the Map data object reference in the Structure View by clicking on it once. Remove it from the presentation by:

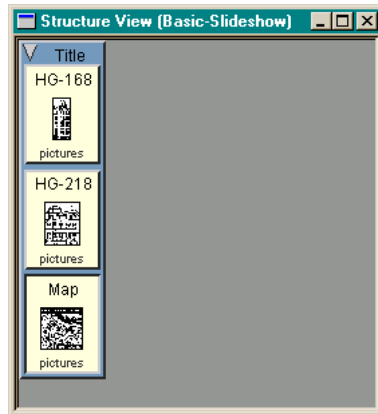
- issuing the Cut command from the Edit menu,
- selecting Cut from under the right mouse button, or
- using the shortcut Ctrl+X. This results in:



Next, select the object labelled *HG-218* by clicking on it once. Now, go to the Edit menu and select Paste, as shown below:



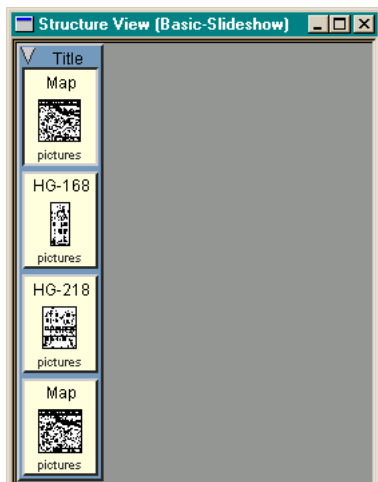
This puts the contents of the clipboard after the item you selected, resulting in:



You can once again see the presentation by selecting Play.

Expanding the Presentation by Using Copy/Paste

As a last exercise in this tutorial, make a copy of Map (now the last object) and place it at the front of the presentation. To do this:



1. select the Map object and then select Copy from the Edit menu
2. select the HG-168 object once
3. paste the Map object selecting the sequence Edit->Paste Special -> Before from the Edit menu

(You could also have selected Copy and then Paste Before using the shortcuts under the right mouse button.)

The resulting Structure View is shown at left. By selecting Play, you can view the work just finished.

Closing Comments on Tutorial 1

The purpose of this tutorial was to acquaint you with the basic features of the GRiNS 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 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 SMIL presentations easily by manipulating the structure of an application.

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

Tutorial 2: Extending an Existing Presentation

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

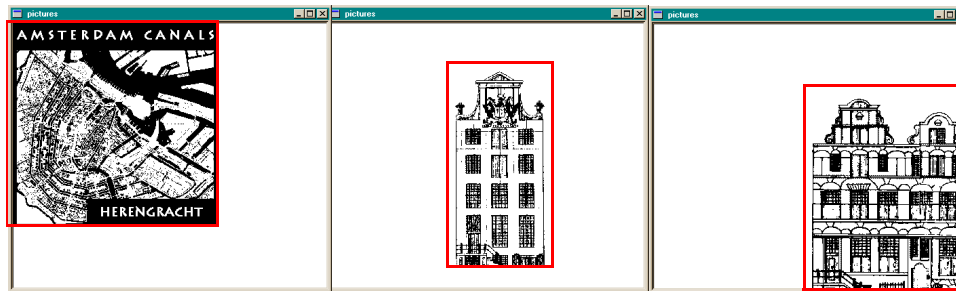
In order to edit this presentation, you:

1. open the editor and select the *Walk.smil* presentation;
2. preview the presentation;
3. learn to navigate thru the structure of the presentation;
4. add some new nodes containing background music;
5. fine-tune the timing relationships between objects using property sheets;
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.

Previewing the Results

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.



(a) use of Top-L Region

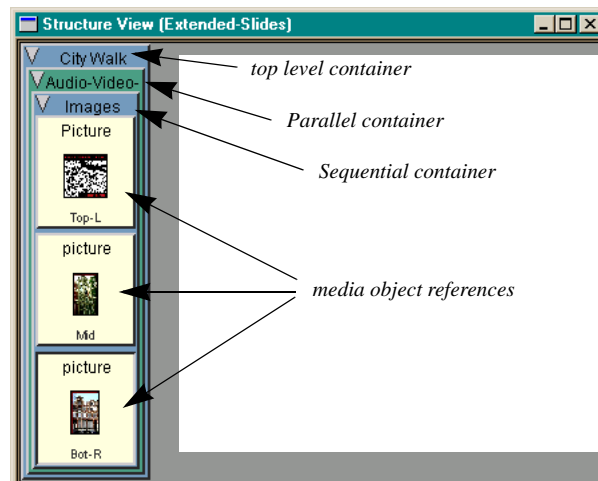
(b) use of Mid Region

(c) use of Bot-R Region

Open the Walk.smil 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.

Start-up the GRiNS Editor and open the *Walk.smil* presentation by selecting Open from the File menu, and browsing the *Examples* 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 essentially the same as that in the first tutorial, with two exceptions:

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

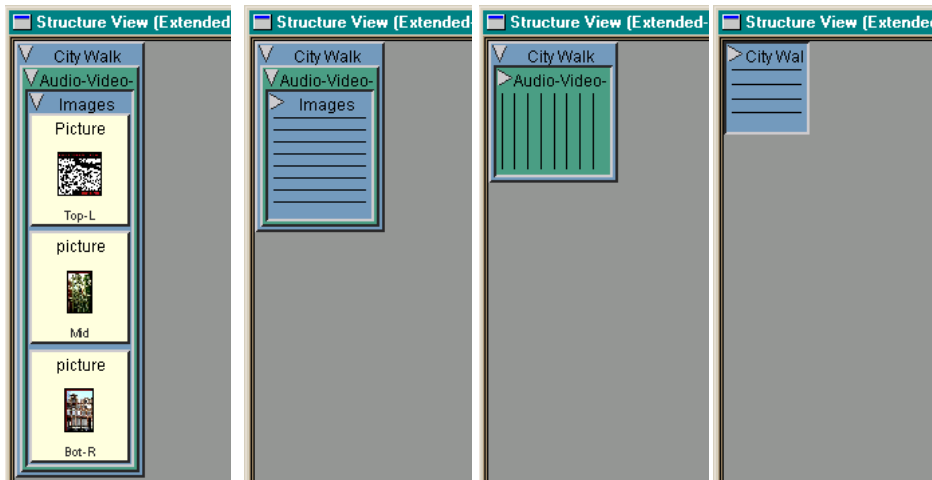
The reasons for the extra structure will become clear in the next section.

As in the first tutorial, we recommend you save a copy of this presentation in the *My GRiNS Tests* folder, this time using the name *TTB.smil* for your file.

Understanding 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-Presentation*. 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:



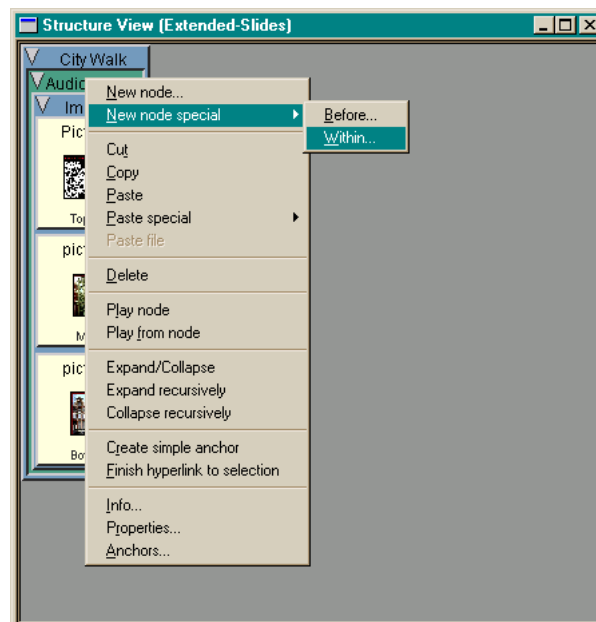
In this simple presentation, opening and closing boxes is not necessary, but this facility is really handy once you start to edit more complex SMIL presentations. But: enough about the future — let's get some work done!

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.

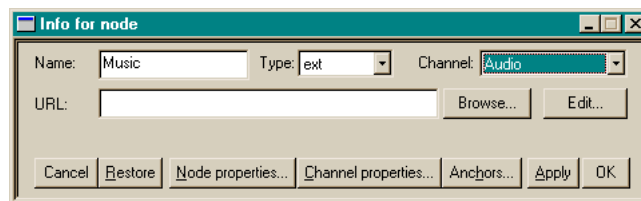
To create a new node, make sure the target structure container is open (by selecting the appropriate triangle), which in our case is the green node labelled *Audio-Visual-Presentation*. Now create a new node by going to the Edit menu and selecting New Node Special -> Within or by using the New Node Special -> Within shortcut under the right mouse button, as shown below:



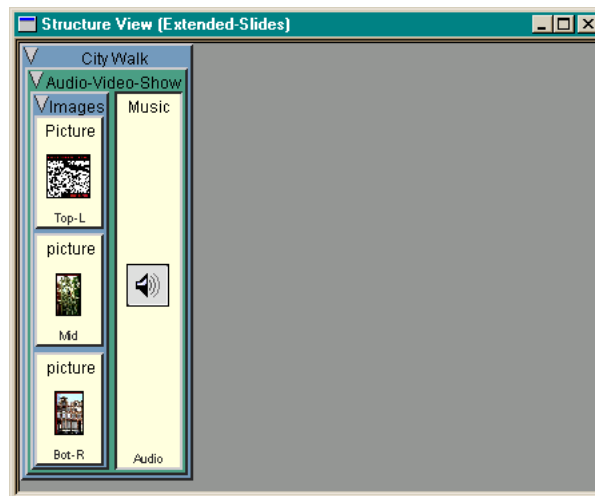
This operation says: in the currently selected structure node (in this case, a parallel node), add a new element within the node. We could have also placed the new node before or after the structure node by using the appropriate commands.

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

- *Name*: Music;
- *Type*: use the selector switch to choose Ext (for external object reference);
- *Channel*: use the selector tab and choose *Audio* as the name in the pop-up box;
- *URL*: you can either use Browse to select the name of the media object, or you can use the drag-and-drop facility to put a data node on the media object reference. (We have a short piece of music in *data\music.aiff*.)



These are all the attributes you need for now, so click OK to commit the entries. In this example, we've left the URL blank and will drag the file *music.aiff* onto the audio icon in the new structure view. The result is:



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 four 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-Video-Presentation" >
    <seq id="Images" >
      <img id="map" ... />
      <img id="HG_168" ... />
      <img id="HG_218" ... />
      <img id="HG_284" ... />
    </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 five 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 eight seconds. This is not very pleasing (having the music be shorter than the images), but it give us something to fix in the next section!

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

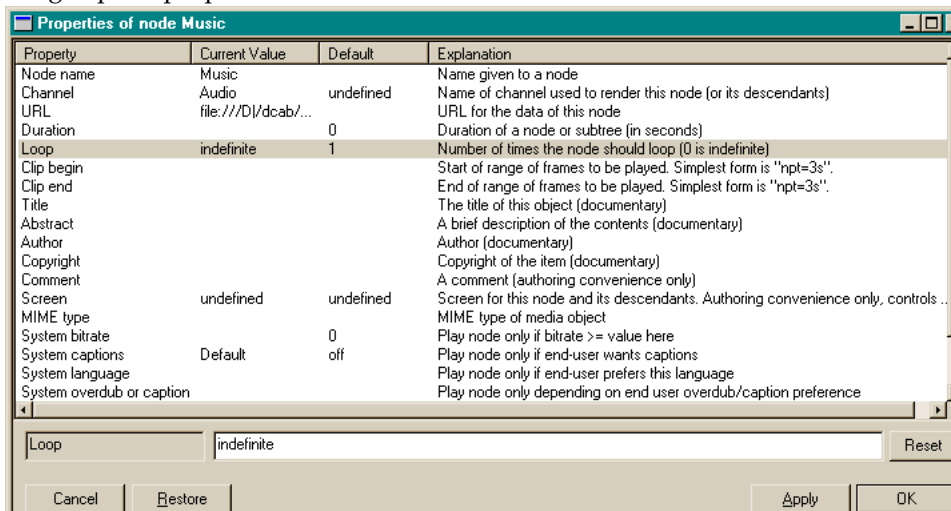
In this section, we show four simple editing operations that will start to develop your SMIL skills for extending the control among objects in a presentation:

1. for objects with a behavior like background music, you can set the loop property, allowing the length of the actual component to be determined by its context in the presentation;
2. in a parallel group, you can explicitly say which object controls the end of all of the members of the group;
3. you can use synchronization arcs in the Timeline view to control relative start times on objects in a structure container, and
4. 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 *audio* 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 *indefinite*, which has the object repeat indefinitely. Confirm with OK. (You can also use the shorthand notation '0', which the editor will map to *indefinitely* — a handy feature for bad spellers...)

Now, preview the new presentation by selecting the:

- Play shortcut bar icon,
- Play menu entry, or
- Ctrl+P shortcut.

The audio will keep repeating until all of the images have been shown -- a total of 15 seconds. Unfortunately, it does more: it keeps repeating until you shut down the presentation (or hit Stop). Our guess is that unless you really like this piece of music, this is not what you want to have in a presentation.

There are several ways of telling GRiNS to do other things with your presentation. First, stop that music!

GRiNS Tip:

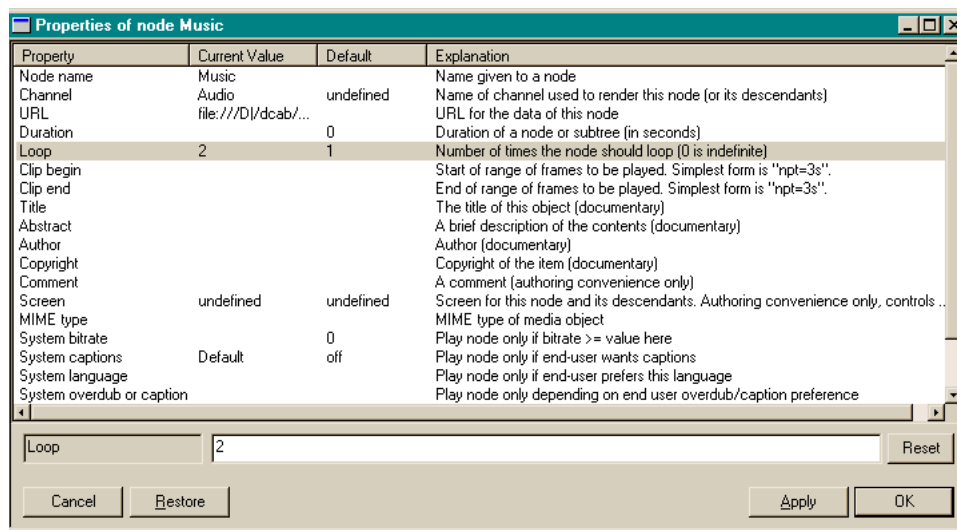
Not all data types can be set to loop. For images or text (or any object that is discrete — that is, that has no 'natural' duration — looping does not make sense. For all continuous, time-based objects, looping makes sense, but not all players support looping functionality. For

Adjusting the loop length

Perhaps the easiest way to change things is to not have the music loop indefinitely, but to give a particular loop value. In our case, you have three pictures of five seconds each, and a piece of music of about 8 seconds. Having the music repeat twice is probably a good starting point to building a more effective presentation.

Use the Tools menu or the right mouse button to bring up the Properties page for the *audio* object. Change the loop counter from *indefinite* to 2. The property page and the associated properties are shown below.

Accept the changes, and play the presentation again. (Note: make sure your Structure View isn't covering the Player view!) You now get to see all of the images and hear the sound twice. As if by magic, both the slideshow and the audio end together.



Setting an explicit duration

Another way of controlling the length of a presentation is to use the Duration attribute. Duration controls the length that an object is active. The duration is not the length of the data element itself (this is determined by the element), but the time that the node in the presentation is active.

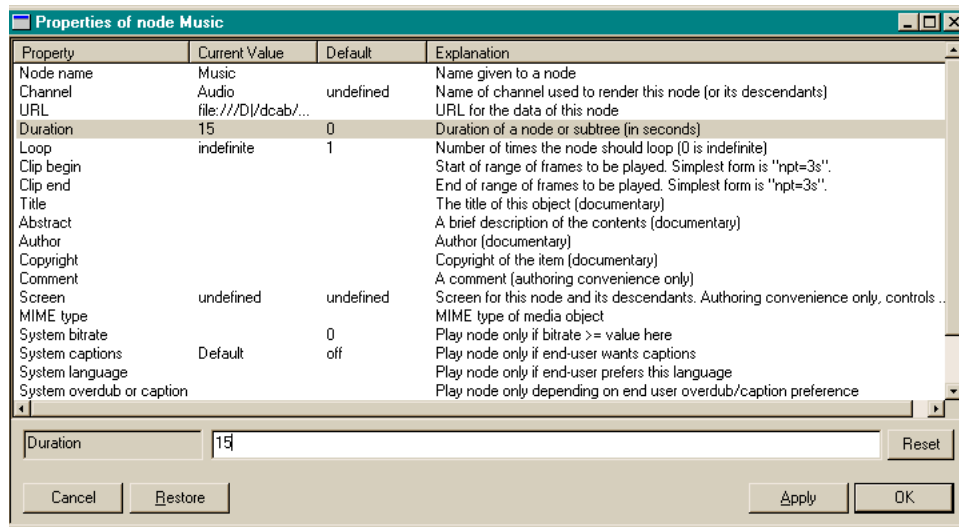
Let's assume you had set the Loop attribute associated with the *music.aiff* file to *indefinite*. As you heard, this means that the object keeps repeating indefinitely.

When you gave it a value of 2, it was rendered twice, for a total of about fifteen seconds. Now, try this:

1. set the Loop back to a value of *indefinite*
2. set the Duration to 15 (or fifteen seconds)

(This is all done in the Properties page for this node.)

The setting look like this:



What happens: the images play in sequence, and the audio plays for the duration you have defined (15 seconds).

So: SMIL lets you set the duration directly via Duration, or indirectly via the Loop counter. In case of a conflict between the Loop and Duration properties, the Duration will always be used to determine the length of the presentation.

SMIL give 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 have defined a Duration of 15 seconds for the audio and a Loop counter of 0 (for indefinite looping). Now, edit the properties of the second image (the one labeled *HG-168*). Change the duration to something like 9 seconds.

The property sheet containing the new duration is at right.

Now play the entire presentation again and determine the result: the first image will appear for five seconds, the second for nine seconds and the third for five seconds.

Unfortunately, the only the first 15 seconds of the entire 19 second period is covered by the audio.

Property	Current Value	Default	Explanation
Node name	HG-168		Name given to a node
Channel	Mid	undefined	Name of channel used to render this node
URL	file:///D:/dcab/...		URL for the data of this node
Duration	9	0	Duration of a node or subtree (in seconds)
Loop		1	Number of times the node should loop (0 for indefinite)
Title			The title of this object (documentary)
Abstract			A brief description of the contents (documentary)
Author			Author (documentary)
Copyright			Copyright of the item (documentary)
Comment			A comment (authoring convenience only)
Screen	Default	undefined	Screen for this node and its descendants
MIME type			MIME type of media object
System bitrate		0	Play node only if bitrate >= value here
System captions	Default	off	Play node only if end-user wants caption
System language			Play node only if end-user prefers this language
System overdub or caption			Play node only depending on end-user option
System required			Play node only if player supports these features
System screen size		0 0	Play node only if screen bigger than this size

Duration: 9

Cancel Restore

There is an easy way to get the behavior we want using the GRiNS Editor. This is shown in the next section.

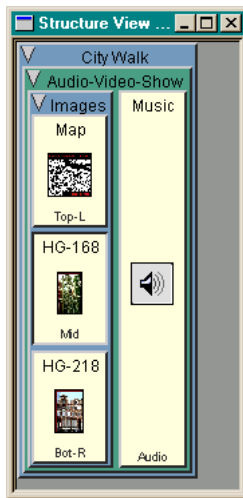
Explicitly Selecting the End of a Parallel Group

Let's think about what you really want this presentation to do:

1. you want a sequence of slides, each of which last the duration defined on the individual attribute pages for the images; and
2. you want the background music to last as long as the total time of all the images, but you don't want to calculate all sorts of time offsets by hand.

If the background music happens to be longer than the sum of the images times, it should be truncated, and if it is shorter, then it should loop until the companion nodes are done.

It turns out that this is easy to do with GRiNS. Rather than changing properties on individual data objects, you want to edit the properties of the parallel structure container that holds both the sequence of images and the audio node.



To locate this parallel container, open the Structure View. You will see the structure at left. Select the structure node named *Audio-Video-Show*. This container has two elements: the *audio* object and a sequential container named *Images* holding the three images.

What you want is to keep looping the audio node until the last element in the sequential node is finished. In other words, *audio* terminates when *Images* completes.

We can accomplish this with the Terminator Child property. While this sounds ominous, it is simply a way of identifying the data or structure node whose temporal duration will determine the length of a parallel group.

To set this property, select the *Audio-Video-Show* node and pull up its properties page using the Properties sheet. Using the arrow keys or the mouse, highlight the Terminator child property, and bring out the pull-down menu in the attribute editing area at the lower side of the property page.

Set the Terminator child to the structure node named *Images*. (The GRiNS Editor keeps a list of valid candidate nodes handy for you.) Now, when the presentation is played, this parallel group will end whenever the *Images* structure node ends. Note that if you later add or remove elements from the Image Sequence node, the length of the audio node will be adjusted automatically.

The sheet before the change is accepted is shown.

Properties of node Audio-Video-Show			
Property	Current Value	Default	Explanation
Node name	Audio-Video-Sh...		Name given to a node
Channel	Default	undefined	Name of channel used to render this node (or its descendants)
Duration		0	Duration of a node or subtree (in seconds)
Loop		1	Number of times the node should loop (0 is indefinite)
Title			The title of this object (documentary)
Abstract			A brief description of the contents (documentary)
Author			Author (documentary)
Copyright			Copyright of the item (documentary)
Comment			A comment (authoring convenience only)
Screen	Default	undefined	Screen for this node and its descendants. Authoring convenience only.
System bitrate		0	Play node only if bitrate >= value here
System captions	Default	off	Play node only if end-user wants captions
System language			Play node only if end-user prefers this language
System overdub or caption			Play node only depending on end user overdub/caption preference
System required			Play node only if player supports these features
System screen size		0 0	Play node only if screen bigger than this
System screen depth		0	Play node only if screen has more bits than this

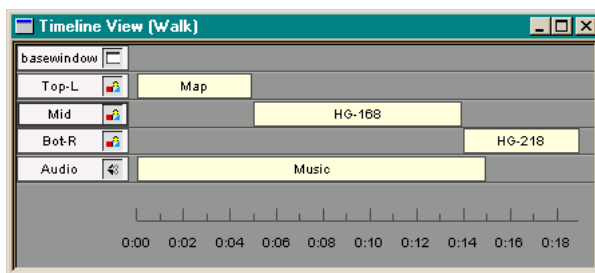
You should try other combinations as well. Note that if the Loop value of any node is set to indefinite, and if that node is set to be the terminator child, then the presentation will keep looping on that node until you hit the Stop key.

Use of the Terminator Child attribute gives you a general mechanism for controlling the duration of (parts of) a presentation. This attribute only makes sense of green parallel structure nodes, however. In a (blue) sequential node, the element that controls the end of the sequential structure is always the last element. Since only one element is active at any time inside of a single sequential container, you never need to choose the closing element.

Note that other attributes (such as Duration) can be set in a sequential node. Look at the property sheet for a sequential node to see the options available.

Introduction to the Timeline View

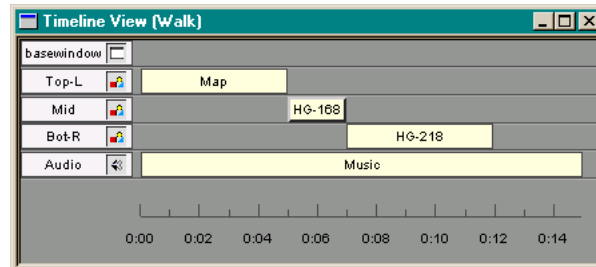
The Structure view shows the logical grouping of elements in a presentation. At runtime, the structural view is mapped to a presentation timeline. An indication of this timeline can be seen by looking at the Timeline View of a presentation. You can bring up this view by selecting it from the Views menu.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



This view shows the *audio* object and *Map*, *HG-168*, and *HG-218*. If you select *HG-168* by clicking on it once, you can bring up its properties via the right mouse button. Find the *Duration* property and change it from 9 to 2 seconds.

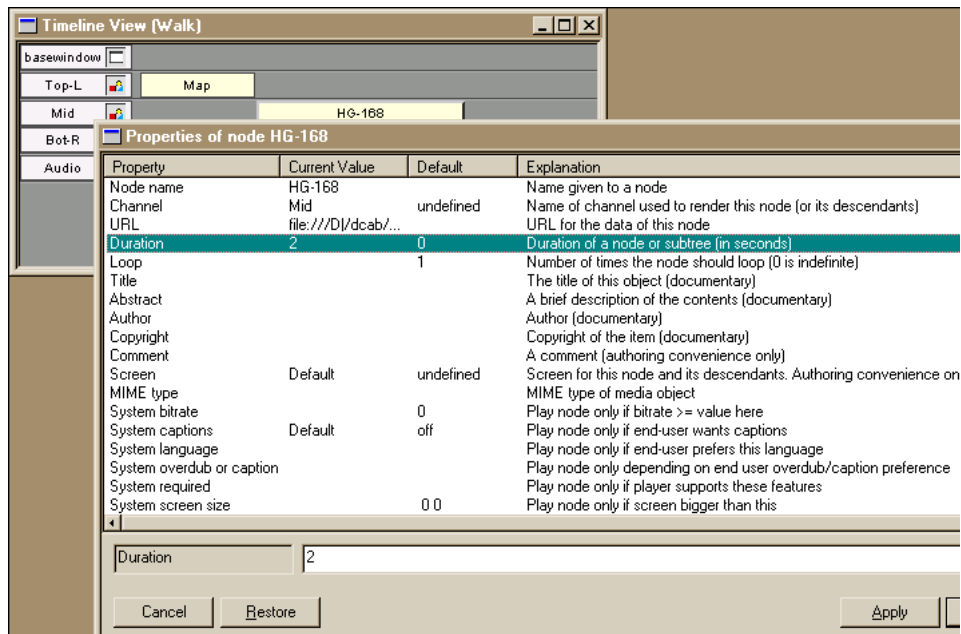
The property sheet for this node (shown on the next page) contains all of the SMIL and GRiNS-related properties associated with this node. The use of most of these properties will be covered in other tutorials.

The result of this is that the length of the *HG-168* object is now scaled to two seconds automatically. The runtime behavior can be confirmed by previewing the node or the entire presentation using the Play options.



Notes:

1. The Timeline displays the 'natural' duration of objects. In the example above, the natural length of the audio object is displayed, rather than the effective length, which is the length provided by the runtime environment. This is done based on author feedback, since most authors want to see how long the building blocks are they have to work with.

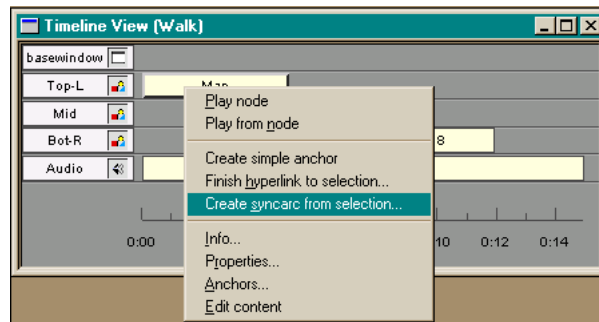


Fine-Tuning via Synchronization Arcs (Sync Arcs)

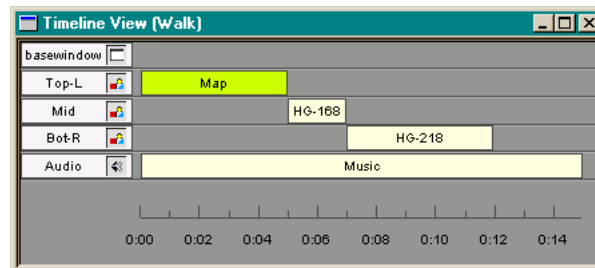
Let's suppose that you wanted to take our presentation-under-construction and make some fine-grained adjustments to timing. For example, let us assume that you want the audio fragment to begin slightly later than the first image. Perhaps you are doing this to balance the delivery load of the presentation,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

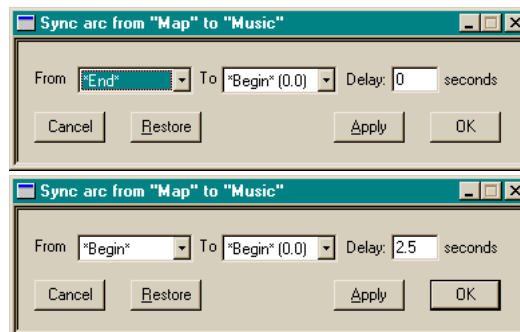
To create a sync arc, select a node in the timeline view. For example, select the picture of the map (named Map) and then, using either the right mouse menu or the Linking menu, select the create sync arc from location command:



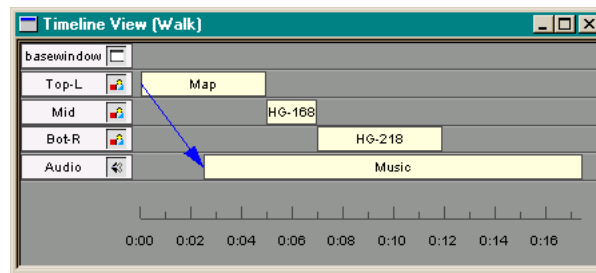
This will define the source part of a sync arc. The source is colored amber to highlight it in the Timeline View.



Once the amber node is shown, select the node named Audio. This will bring up the Sync arc properties box, as shown top right. 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.



Once the sync arc is accepted, the arc is drawn as a blue arrow between the start of *MAP* and the start of *Audio*. 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.

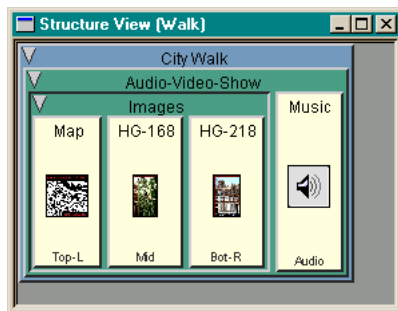
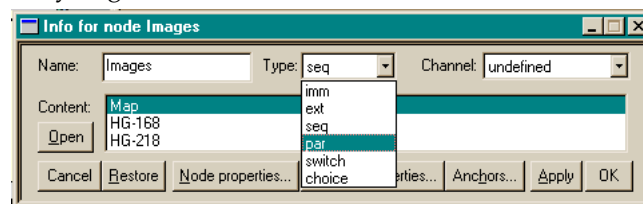
Notes:

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.

Better Understanding 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, select the Images structure node and bring up the Info 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* 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 *picture_2* for three seconds.

The audio lasts as long as the longest image (instead of the sum of the image times).

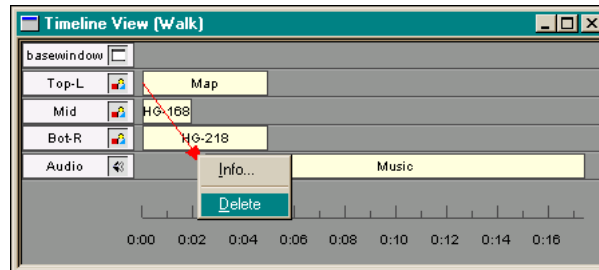
A copy of the screen at the start of the presentation is shown on the next page.



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, by the music starts 2.5 seconds after the Map has been shown. This behavior can best be seen by looking at the timeline view. Here you see the images all starting in parallel, with the audio starting offset from the common start times of all of the images. (See next page.)

The timeline view can be used for a variety of time-related functions with GRiNS. We revisit it in Tutorial 4.

In order to finish up this tutorial, reselect the Structure view of the presentation and turn the re-structured Image Sequence back into a sequential structure node. You should also delete the sync arc, since you won't be needing it in this example.

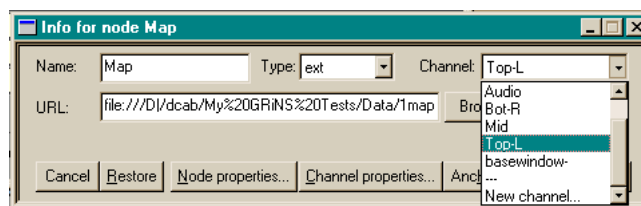


Selecting Layout Regions

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 defeat resources available, as defined in the SMIL standard. In the second tutorial, you used four Resource Channels that were defined for us in the template.

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

Open the Structure View, and select the first element in the *Images* — this is the *MAP* object. Next, bring up the Info box 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 Info box 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

The purpose of this tutorial is to create the *Walk.smil* SMIL presentation that was used in Tutorial 2. The *Walk.smil* template 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.smil* 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.

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. (For more information, see the *GRiNS User's Guide* and the *GRiNS Reference Manual*.) 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.smil presentation

In the first version of the Slideshow in Tutorial 1, we made use of the default rendering regions available with any SMIL 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.smil 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 the Walk.smil presentation, each media object is placed on a separate channel. This is acceptable, 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 examples below.

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 or inches). Each of the SMIL Regions (if they are used) are placed inside the presentation's base window. Each Region is defined by an X, Y coordinate within the base window, followed by either an absolute size (in pixels, mm or inches) 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.

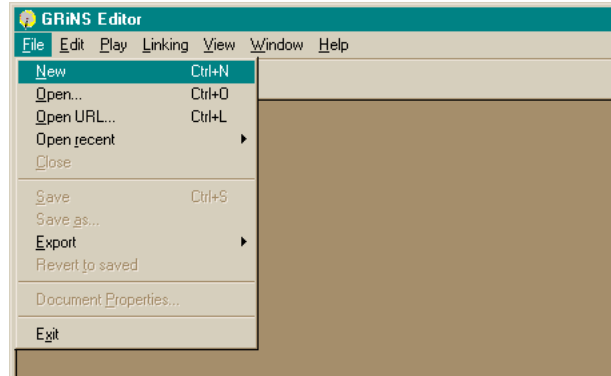
You specify a SMIL Base Window and the individual SMIL Regions via (Resource) Channels in GRiNS. The Channels provide a container for basic layout attributes and they provide an extra measure of control by allowing you to assign media types to individual SMIL Regions. This can be used to make playback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

The GRiNS Editor also allows you to define *Layout Groups* (or simply Layouts); Layouts 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.

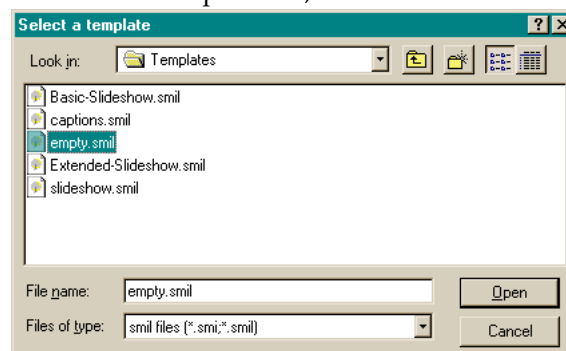
In this simple presentation, we will ignore the concept of a Layouts. Instead, we make the decision to define three regions on the screen for displaying three images in sequence in different places.

Open the GRiNS Editor and select the EMPTY.SMIL template

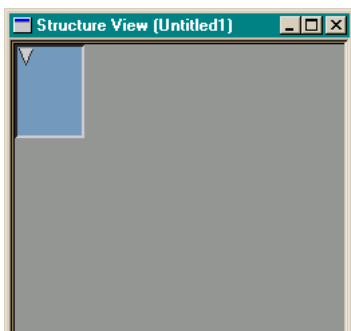
If you have not already done so, start-up the GRiNS Editor; go to the File Menu and select New.



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



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



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

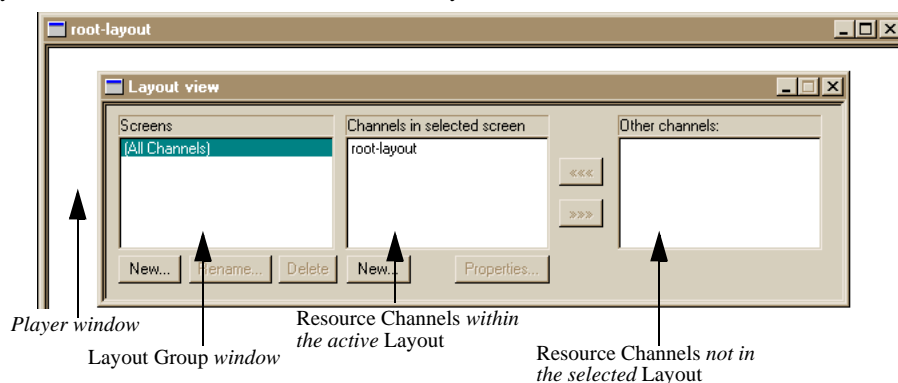
Once the *Empty.smil* template is opened, you can start with the process of designing your application (or application template). There are many ways of building a SMIL presentation. You can take a *data-centric* view by first defining all of the objects in the presentation and their

relative order or activation, or you can take a *presentation-centric* view by first designing your layout region(s), and then later assigning object 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.

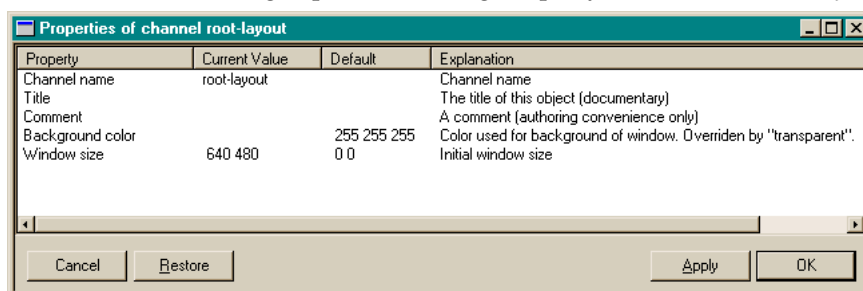
For the sake of simplicity, this tutorial takes the latter view: we start with designing a simple layout space and then identify a number of objects to be presented.

Define Regions for Displaying Presentation Components

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



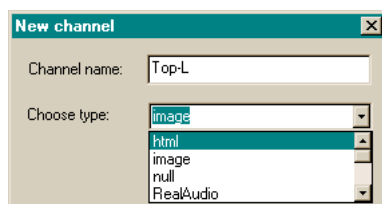
GRiNS provides you with a *root-layout*, which is the rendering canvas for the presentation. The default size of this canvas is 640 x 480 pixels. You can resize the window using the standard window resizing techniques for your computer (such as dragging the lower-right corner up or down, or left or right), or you can edit the various properties associated with the *root-layout* by selecting the name *root-layout* in the middle (Channel) window of the Layout View, and then selecting the Properties button. This brings up the following Property Sheet for the *root-layout*:



The main elements to alter are the name, the background color of the window and the window sizes. We don't need to alter these defaults now, so close the window to expose the Layout view window.

Define SMIL Regions via GRiNS Resource Channels

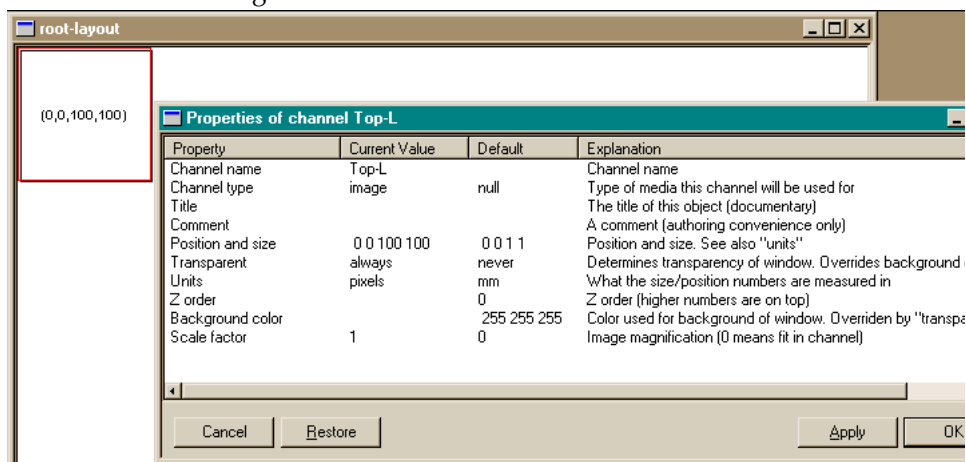
A SMIL presentation allows you to put any number of regions inside a presentation root layout. Find the section labelled Channels in the middle of the Layout view, and then select the New... button. This will bring up a dialog box that allows you to define the Channel.



A Channel consists of a name and a type. In GRiNS/SMIL, the following channel types are supported:

- HTML: a channel in which HTML-formatted documents are rendered;
- Image: a channel in which various types of image formats are supported;
- Layout: the GRiNS name of the SMIL Base Window;
- Null: a channel without any particular resource policy;
- RealAudio, RealPix, RealText and RealVideo: channels that allow you to integrate RealNetworks' RealMedia objects in a presentation;
- Sound: an audio channel.

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 by double-clicking on it in the Channels list. The property sheet contains the following elements:



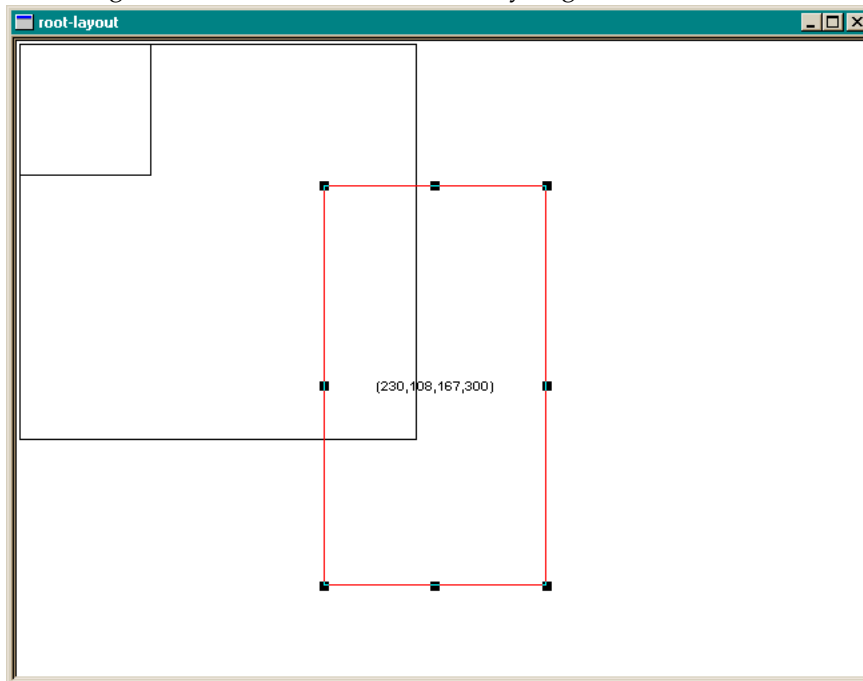
(The default window is shown at top left, and the new window parameters are shown in the property sheet.) You can change the location of the window and its size by entering the parameters directly into the property sheet. This is useful if you want pixel-level control over your application.

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).

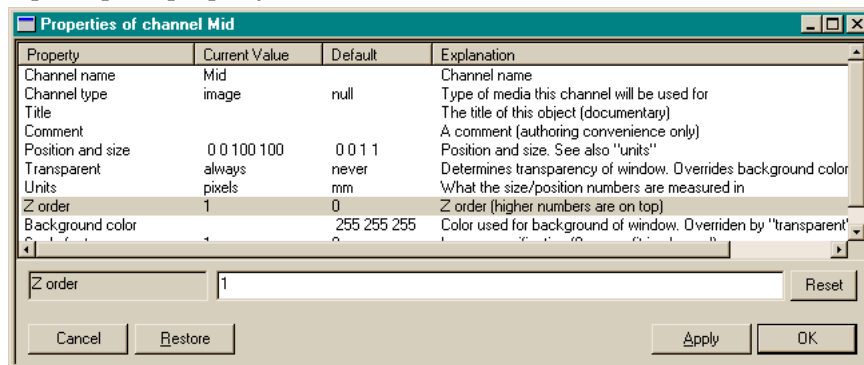
The following illustration shows the view that you get:



Note that the original position remains displayed for reference until you click somewhere outside of the Channel.

Once you are done with the positioning, you can accept the changes by clicking anywhere outside the channel..

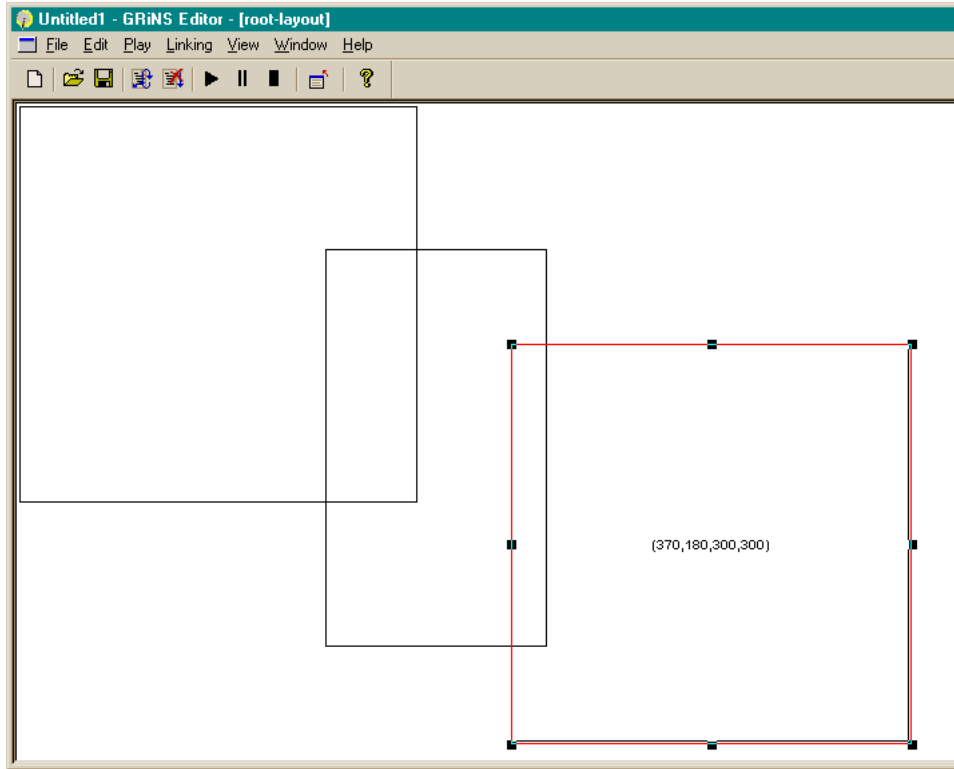
Now, pull up the property sheet for *Mid*, as shown below.



Go to the *Z order* 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 two choices that we now consider: 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.

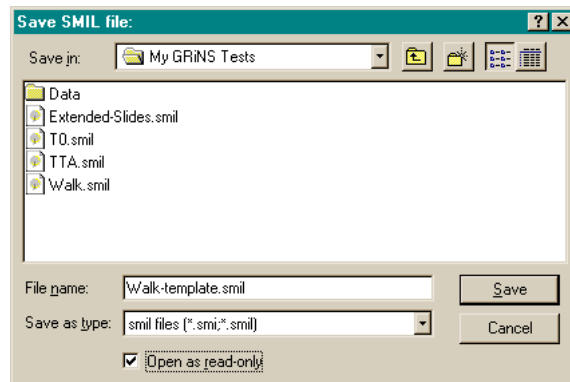
You can now go on to create the third channel by selecting New from under the Channels list. Give the Channel the name *Bot-R* (bottom right) and give it a type of *Image*. Using either the property sheet approach or the resizing via the mouse, place the Bot-R channel in the bottom right part of the 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 the mouse-based resizing, we would get the following view:



At this point the presentation 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.

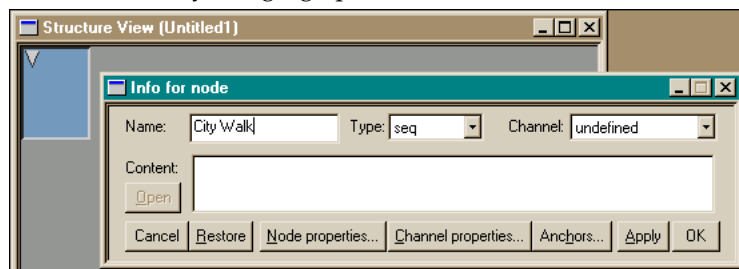
If you haven't done so already, this is a good time to save your work. Use the name *Walk-template.smil*, and select the 'open as read-only' check box to create a template. The save box looks as follows:



Defining the Structure and Contents of the Presentation

The *Walk.smil* 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 Info box. Inside, we label the



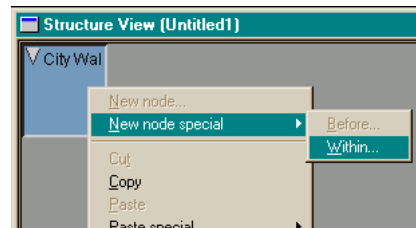
structure node *Extended Slideshow*. The other properties are already filled in by the Editor.

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 New -> Within from under the right mouse button.

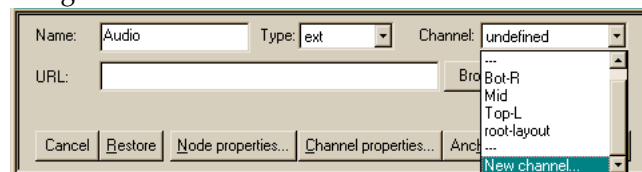
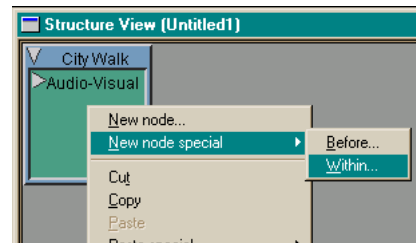
The GRiNS Editor is aware that we only have one option here: you can't put anything before or after the top-level container, so the only choice is to put a new node inside (or within) that container.



The type of node we are about to create is a parallel structure container, which will hold the audio and images in the presentation. Once the node has been created, select its Info box and name the node *Audio Visual Show*. In the Info box, you should also set the node type to be 'PAR', for a parallel container.

We now continue building the presentation by placing a new node inside the *Audio Visual show*. We again select New Node > Within, but this time we create a media object reference instead of a structure node. Name the node *Audio*, sets its type to *ext* (for external) and select *New Channel* under the channel window.

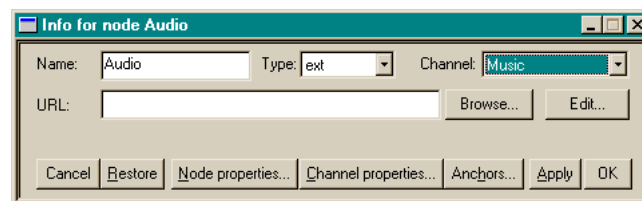
The filled-in dialog box is:



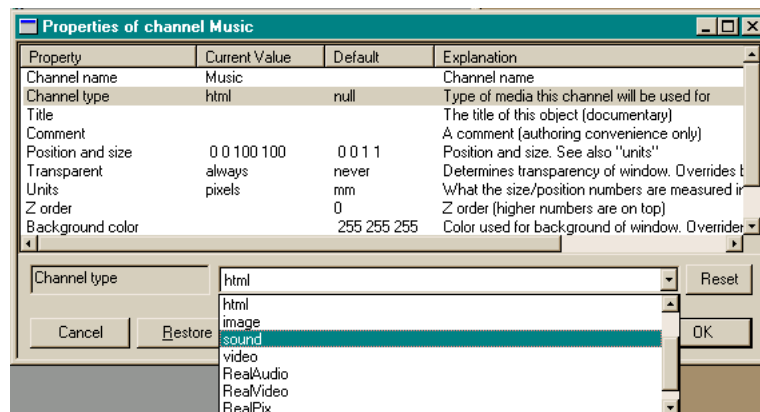
When you select New Channel, you will be given a *Channel name* pop-up box. You should name the new channel *Audio*:



The resulting Info box is:



Recall that we haven't defined a layout region for Music, but by creating a new channel named Music, GRiNS has given this channel some default properties. The default media type for this new channel is HTML. This is probably not what we want for an audio channel, so it is probably a good idea to define the properties that we actually want for the Music channel. We do this by opening the channel's properties, using the Channel properties button at the bottom of the Info box. You get:



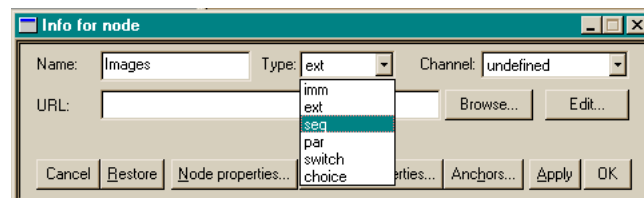
After creating the audio node, the Structure view will look as shown at left. What we have defined so far is a presentation structure with a single audio media reference.



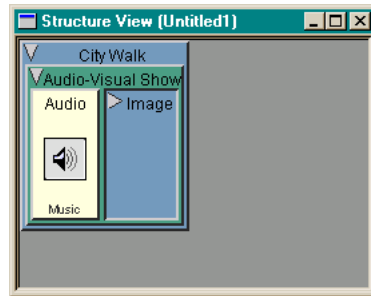
We now want to define the sequence of images in parallel with this object.

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 Audio node and then select New 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 Audio object.

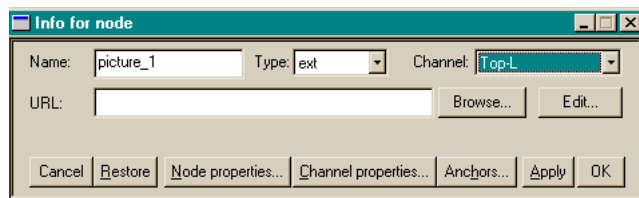
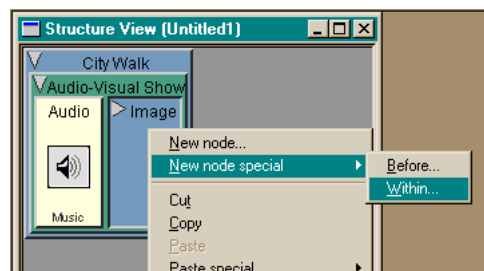
Once the new node is created, open that node's Info box and define the high-level characteristics of the node. Name it *Images*, give it a type of *seq* (for a sequential node) and leave the channel type undefined. The resulting Info box is:



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. We start by defining a new node within the Image Sequence container:

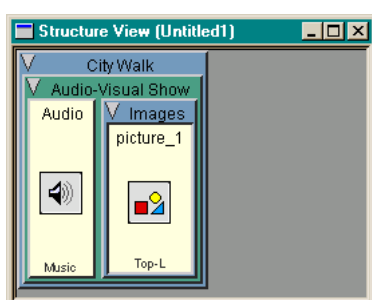


We open the Info box associated with this node, and define an image reference. The name is *picture 1*, the type is *ext* (for external) and the channel that we

will use to display this image is *Top-L*, which is one of the channels we defined earlier in this tutorial.

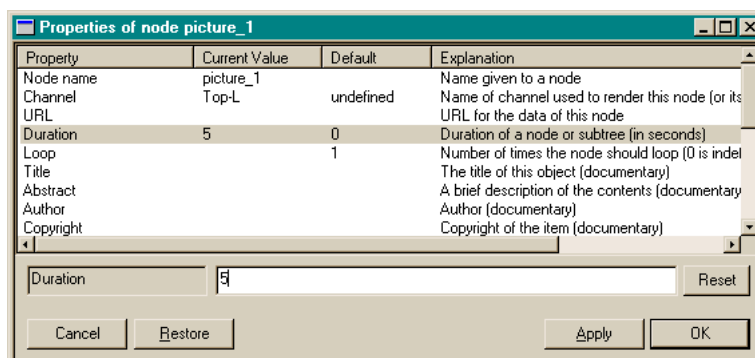
The node is defined as *picture 1*. This is not the name of an actual file (or network URL), but simply a text line that will tell the user of the template what kind of

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.



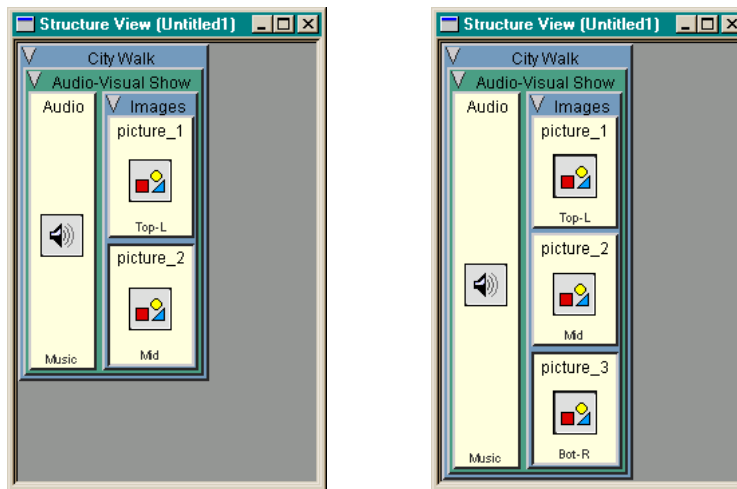
Once you accept the information in the Info box, a media object reference will be created and the Structure view will be updated with a tan block named *picture 1*, and containing an image icon. (If an actual file name was used, the object reference box would contain a thumbnail image of the referenced file.) At the bottom of the object reference is the name of the channel on which this image will be rendered (in this case, Top-L).

IMPORTANT: what you have created is a place-holder 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 place-holder. 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 Info boxes for each node and change the node name, the channel assignment (Mid for *Picture 2* and Bot-R for *Picture 3*), and the name of the file/URL to be referenced.

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.

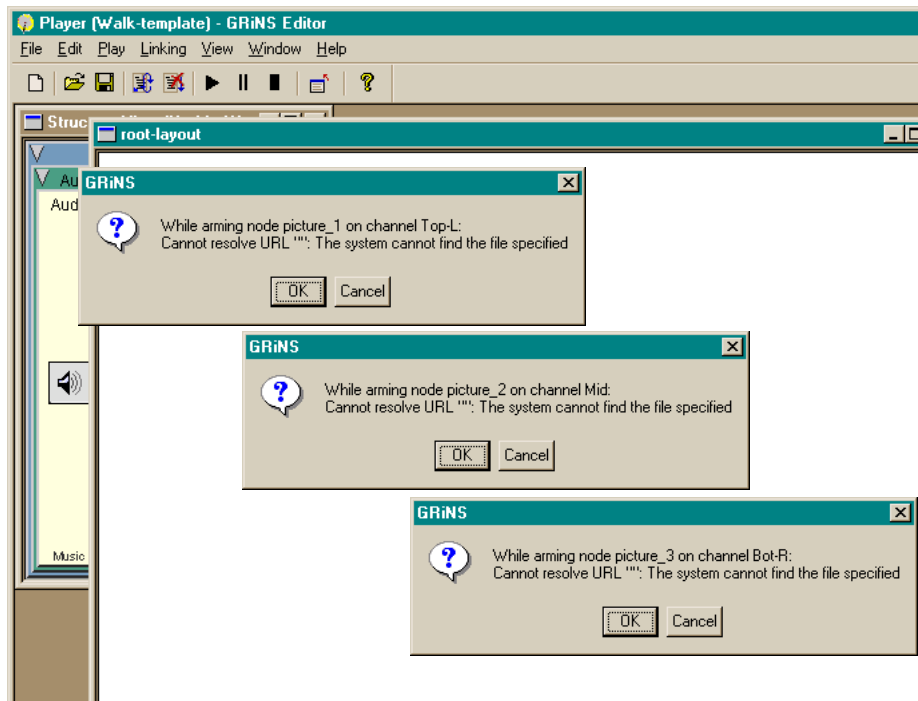
Previewing the Presentation

The Walk-template.smil 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 place-holder 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 show 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 bit of 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 4.

Tutorial 4: Working With Anchors, Links and Hypermedia

This tutorial is being revised and will be available shortly.

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

This tutorial is being revised and will be available shortly.

GRiNS Quick Reference Information

SMIL Compliance Information

The GRiNS Editor for SMIL and the GRiNS Player support 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 supported in a future release.

Supported Media Table

The following chart gives a listing of the media types supported by various versions of the GRiNS Editor and Player for SMIL.

Table 1:

MIME type	Extensions	Windows 95/98/NT	Mac	SGI Irix-6.3	Sun Solaris-2.5
audio/32kadpcm		no	no	no	no
audio/basic	au	yes	yes	yes	yes
audio/x-aiff	aiff, aifc, aif	yes	yes	yes	yes
audio/x-wav	wav	yes	yes (1)	yes (1)	yes (1)

Table 1:

MIME type	Extensions	Windows 95/98/NT	Mac	SGI Irix-6.3	Sun Solaris-2.5
image/cgm		no	no	no	no
image/g3fax		no	no	no	no
image/ief		no	no	no	no
image/jpeg	jpeg, jpg	yes	yes	yes	yes
image/naplps		no	no	no	no
image/png	png	yes (2)	yes	yes	yes
image/tiff	tiff, tif	yes	yes	yes	yes
image/x-portable-any-map	pnm	no	yes	yes	yes
image/x-portable-bitmap	pbm	no	yes	yes	yes
image/x-portable-gray-map	pgm	no	yes	yes	yes
image/x-portable-pixmap	ppm	no	yes	yes	yes
image/x-rgb	rgb	yes	yes	yes	yes
image/x-xbitmap	xbm	no	yes	yes	yes
image/x-xpixmap	xpm	no	no	no	no
image/x-xwindowdump	xwd	yes	no	no	no
	bmp	yes	yes	yes	yes
	fts	no	no	no	no

Table 1:

MIME type	Extensions	Windows 95/98/NT	Mac	SGI Irix-6.3	Sun Solaris-2.5
	pm	no	no	no	no
	ras	yes	no	no	no
	tga	yes	no	no	no
video/mpeg	mpeg, mpg	yes	yes	yes	yes(6)
video/quicktime	qt	yes	yes	yes	no
video/x-msvideo	avi	yes	yes(5)	yes(5)	no
video/x-sgi-movie	mov	no	no	yes	no
text/html	html, htm	yes (7)	yes (3)	yes (4)	yes (4)
text/plain	txt	yes	yes	yes	yes

Notes

1. Uncompressed WAV only.
2. Support seems to be somewhat buggy.
3. Very limited.
4. HTML 2.0 only.
5. Not all encodings supported.
6. With Quicktime 3 or QT MPEG extension installed
7. Both IE4/5 and WebsterPro controls supported

Support for RealMedia (G2) Data Types

The following chart describes the levels of support provided for the listed RealMedia data types used in the RealNetworks G2 player:

Table 2:

RealMedia type	Extension	GRiNS Editor		GRiNS Player	
		Recognized	Rendered	Recognized	Rendered
RealAudio	ra, rm	yes	yes	yes	yes
RealVideo	rv, rm	yes	yes	yes	yes
RealText	rt	yes	yes	yes	yes
RealPix	rp	yes	yes	yes	yes
RealFlash	rf	yes	yes	yes	yes

Notes

1. *Recognized* in the Editor means that RealMedia regions can be specified as part of a document. A G2-compatible presentation can be authored. Timing of RealMedia objects will not be displayed correctly.
2. *Recognized* in the Player means that data objects containing RealMedia items will be parsed correctly and will not cause the Player to crash.

Access methods

The GRiNS player supports the following access methods (URL schemes).

- file
- http
- ftp
- data
- gopher

References and Links

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

