GRiNS Tutorial Guide

Editor/Player version 0.5

Windows, Macintosh and UNIX

GRiNS/SMIL Tutorial Guide

Important Notices

This is a pre-release version of the *GRiNS Tutorial Guide* for Version 0.5 of GRiNS/SMIL, dated May 11, 1999. All of the information has been verified, but we cannot guarantee complete consistency of this guide.

─────────────

This version of the *GRiNS Tutorial 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 quality, 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/SMIL 0.5-win32-0 version (for Windows-95|98|NT-4). While the look of other versions of GRiNS are slightly different because of common conventions on those other environments, the functionality described is similar for all versions. Only the Windows-related images have been include to reduce download time.

─────────────

We welcome your questions on GRiNS/SMIL and comments on this documentation. Please submit all questions and comments to our support desk at *grins-support@oratrix.com*. We will try to give you a response within one working day. We will also 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 such a server. 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*.

GRiNS/SMIL Tutorial Guide

Table of Contents

GR*i*NS Tutorials Roadmap

Thank you for downloading the GR*i*NS toolset for playing and (optionally) creating SMIL presentations.

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

Distribution Package Contents

The GR*i*NS distribution package consists of the following components:
- *GRiNS/SMIL Quick Start Guide*: an overview of the installation instructions for GR*i*NS∕SMIL and a quick tour of the basics of the GR*i*NS 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 you own presentations;
- *Examples*: a collection of SMIL demonstrations, some of which are referenced in the *Quick Start Guide*;
- *Software:* depending on the distribution you downloaded, a GR*i*NS 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/SMIL Quick Start Guide* before going through the tutorials in this guide.

GRiNS Tutorial Guide

The *GRiNS/SMIL Tutorial 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. *Using Object in Parallel and Audio*: 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 guickly build presentations that can adapt to the environments that your users may encounter.
5. *Hypermedia Support in GRiNS*: Linear presentations are great, until you 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/SMIL User's Guide* and the *GRiNS/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 create SMIL documents using the GRiNS Editor. SMIL is a W3C Recommendation for defining structured multimedia presentations for use across the Internet. Often such use will be in conjunction with streaming multimedia players, such as the RealNetworks G2 player. Of course, you don't need streaming media support to define a SMIL presentation: SMIL gives any SMIL-compliant player all of the information it needs to schedule and present a combination of media elements as a single presentation.

The *GRiNS Editor for SMIL* provides a single interface that allows you to create and export presentations to any SMIL compliant player. GRiNS uses an embedded previewer that helps you define you presentation quickly and correctly, and a series of embedded and optional export tools that will target your presentation for use on particular SMIL players.

The following illustration above gives an overview of what you can do with the current version of the GRiNS Editor for SMIL.
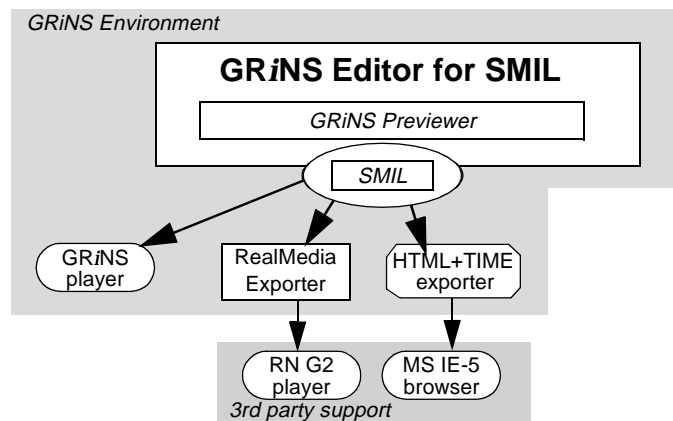
The GRiNS Editor for SMIL consists of an editor for creating, maintaining and previewing SMIL presentations. The environment also consists of a stand-alone SMIL-compliant GRiNS Player and several export tools that allow you to target various industry-standard and prototype players.

GRiNS lets you define a presentation in terms of a variety of industry-standard media formats. The current version includes all popular image, audio and video formats for use on stand-alone systems, as well as HTML and plain text. A description of the supported media types is given at the end of this document.

The tutorials in this guide allow you to build and preview presentations for use with any SMIL-compliant player, such as the GR*i*NS Player, the RealNetworks *G2* player or the Helios *Soja* player. Presentations can also be exported to an HTML+TIME temporal specification. The tools required to define G2 or HTML+TIME-based presentations are available separately and have their own tutorial introductions.

The RealNetworks G2 player is the best known SMIL-compliant player, and the HTML+TIME functionality embedded inside Internet Explorer-5 is a prototype-level release of one approach to placing SMIL-based functionality within HTML. The RealNetworks G2 player is appropriate for users who wish to create streaming media personations based on SMIL, and the HTML+TIME format is appropriate for people who wish to experiment with embedding timing control in HTML documents.

Notes:
1. RealNetworks G2 player often makes use of Real's proprietary data types (RealAudio, RealVideo, RealText and RealPix). Objects in these formats can be included in any presentation constructed via GRiNS, but previewing can only be done via a RealNetworks G2 player.
2. The HTML+TIME format is not based on any accepted standards and is subject to change and evolution.

See *www.oratrix.com/GRiNS* for more details and information on availability of the authoring and presentation management tools available within the GRiNS product suite.

A Note on the Illustrations Used in the Guide

Although GR*i*NS is available for Windows, Macintosh and UNIX platforms, this Quick Start 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 File->Close window closes a single view, File->Close document closes the document (and all views). The Views->Open Document 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.

GRiNS/SMIL Tutorial Guide

Tutorial 1: Creating a Simple Sequential 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:
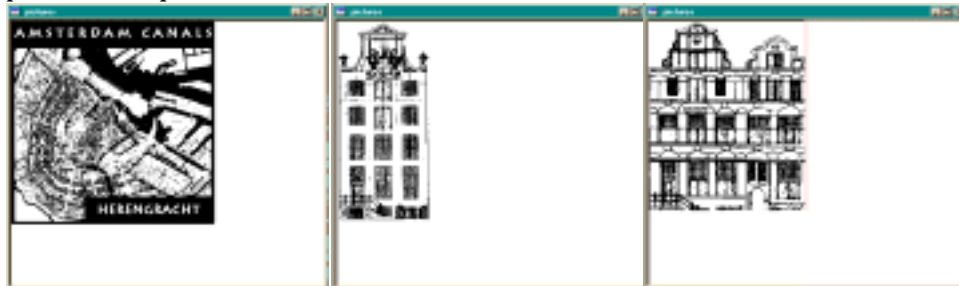
After some modifications, the presentation will show a fourth image, which is a copy of the first.

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:
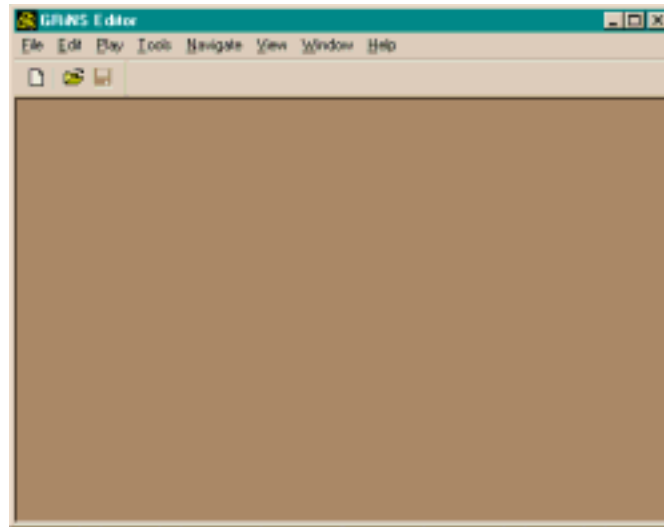
Next, go to the File menu and select New, which allows you to select a GRiNS Template file. In this tutorial, select the *Basic-Slideshow.smil* file, as shown below:
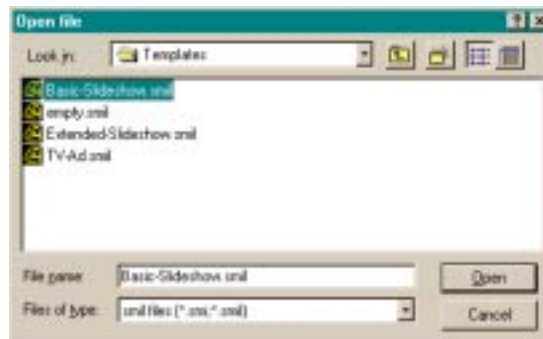
This will open the SMIL document *Basic-Slideshow*, giving you the Editor's basic view:
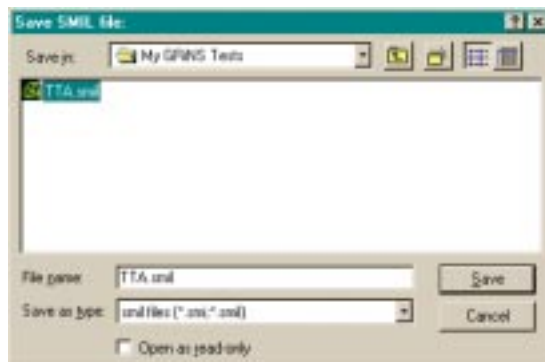
Before doing anything else, it is good practice at this point to immediately save you newly-opened file somewhere. We recommend that you create a folder called *My GRiNS Tests* to save your work-in-progress. (You may also put the file elsewhere, of course.) Select Save as ... from the File and name your file T*TA.smil.*

NOTES:
1. The opening screen assumes that you use the default configuration for GRiNS as distributed;
2. 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;
3. If you work on a multi-user system, make sure you don't overwrite files created by other users of this tutorial.

Editing in the GRiNS Structure View

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 as 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 View menu and selecting Structure View, as shown below:
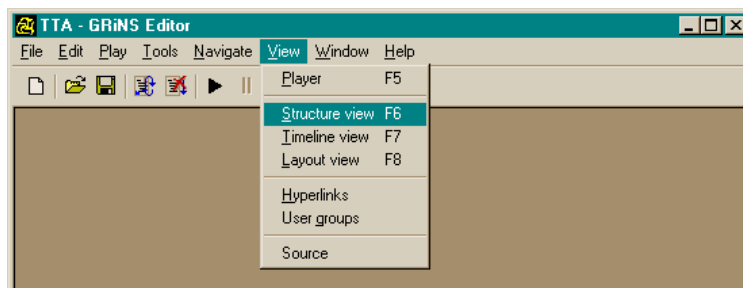
For an introduction to this and other Views within GRiNS, you should consult the GRiNS Quick Start Guide, available from the GRiNS Web site:
www.oratrix.com/GRiNS/Docs.

With respect to this tutorial, the Structure View gives you an overview of the structure of the Basic-Slideshow presentation:
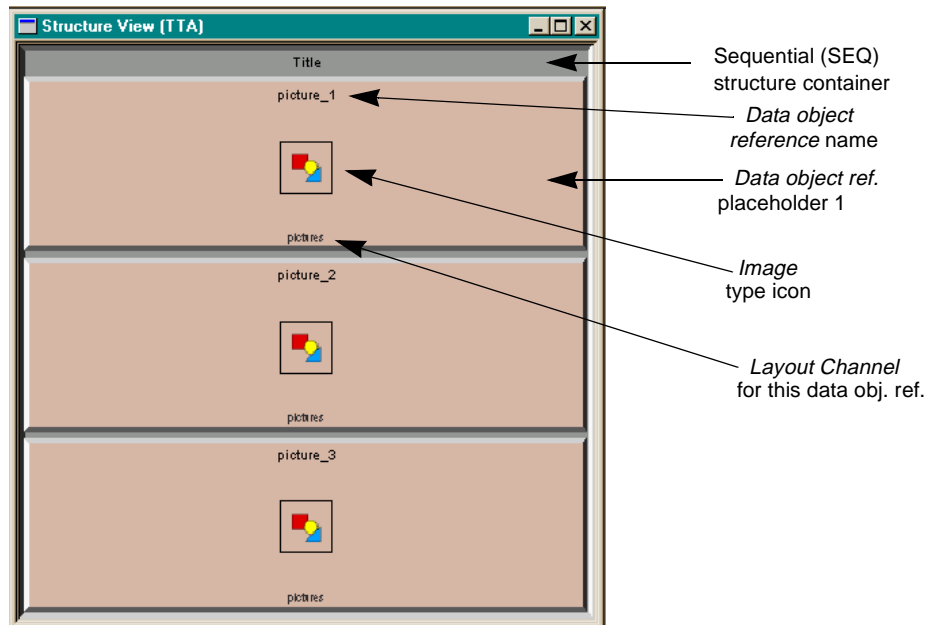
The presentation consists of three data object references (the light-colored boxes) and one structure node (the grey box labelled *Title*). Each of the data object references (will are called data objects), labelled *picture_1*, *picture_2* and *picture_3*, refers to an instance of an image. An image instance 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.

9

Because this is a template file, the structure of the application is complete, but we still need to fill in the individual image file names and locations.

Fill in the First Image for Use in the Show

We are now ready to fill in the blanks in the presentation.

Click on the box labelled *picture_1* once. Doing so will give the following view:

highlighted
data object reference
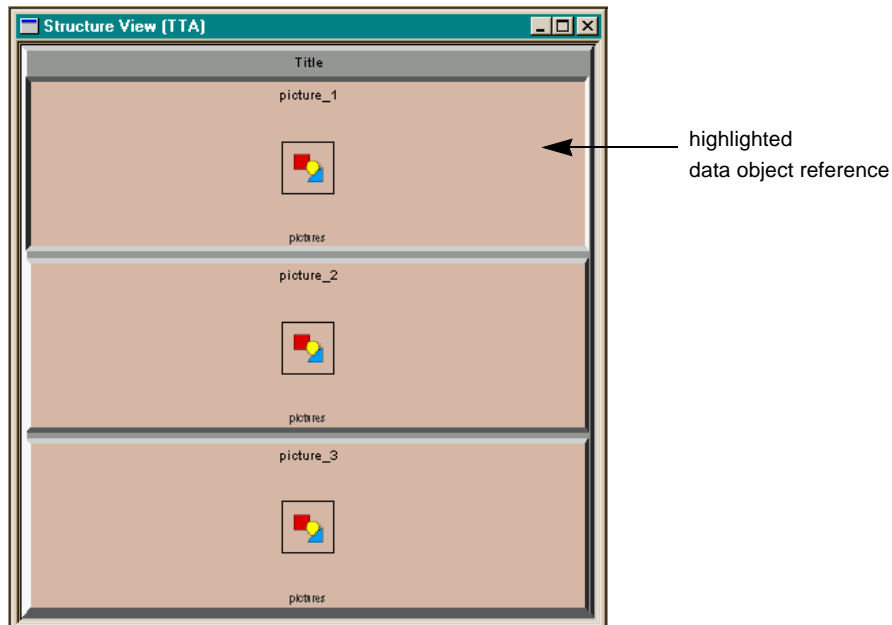
The *picture_1* box has been highlighted. Since this box contains a data object reference, not a data object, we need to define which real image is associated with this reference. To do this, we need to open the box and fill in the name (and possibly location) of the data object we would like to display first.

To do this, we go to the *Tools* menu, and select Info, as shown below:
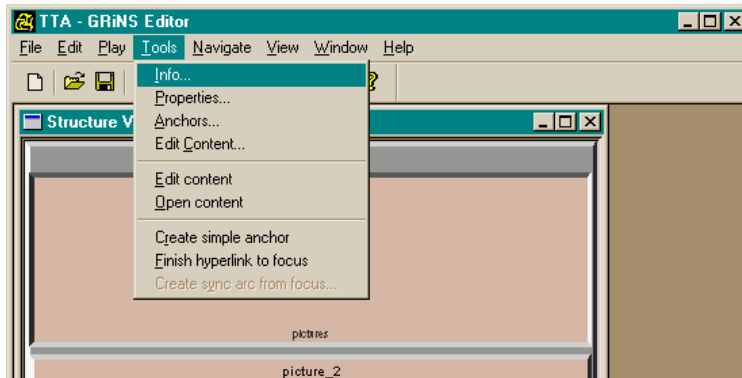
This gives a dialog box where you define the basic properties of the reference:
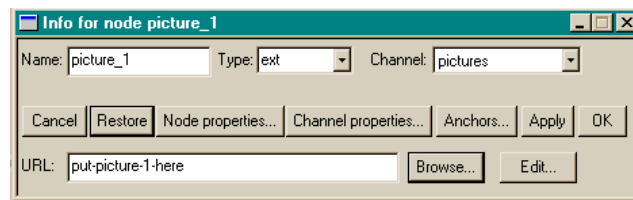
You should do the following:
1. change the name of the node from *picture_1* to *HG-168*. This gives a name to the data object reference, but it does not identify a particular object to be associated with this reference.
2. select the Browse button to select a data object to be associated with this reference. Use the file named *h168.jpg*, which is located in the *Data* folder provided with these tutorials. The result is the following Info box:

11

3. confirm all selections

After filling in the details, the Structure View should look as follows:
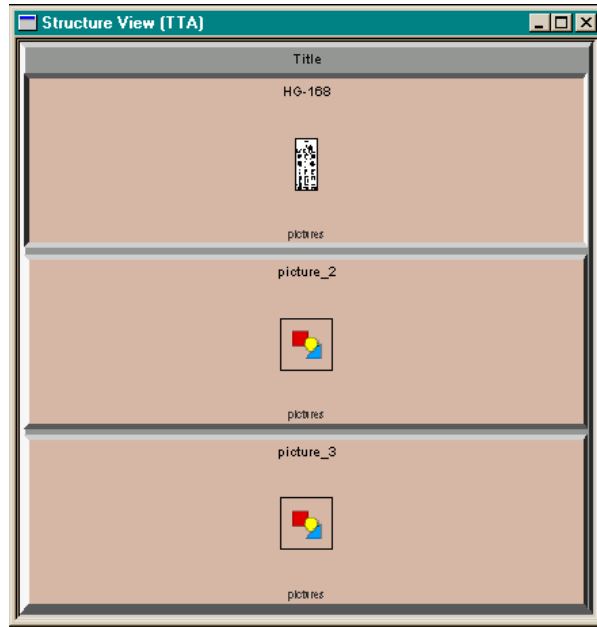
The name of the top data object reference has changed to *HG-168*, a small thumb-nail of the selected image is displayed.

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-218 and set the file to h218.gif 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 MAP and set the file to map.gif.
7. confirm all selections.

The result is the following Structure View:
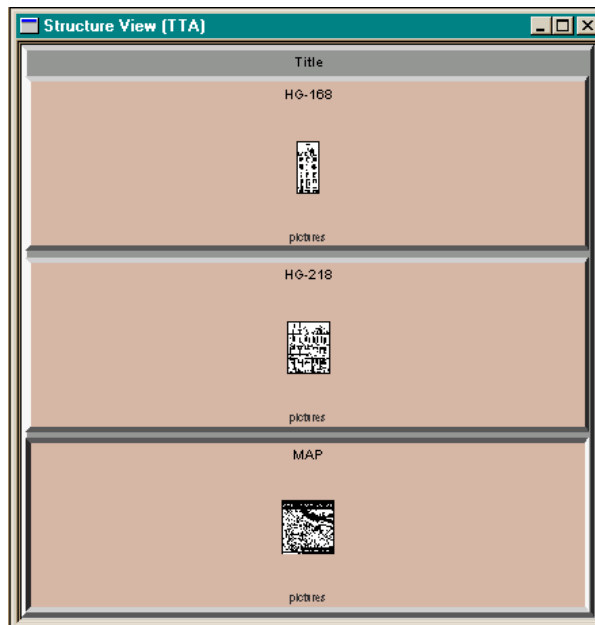
Finally, we should give this structure container a meaningful title. (This is optional, but it provides information that some players can use to index your presentation.) To do this, locate the grey box labelled *Title* at the top of the view and click on it once. The box you have just selected is a sequential structure container -- it holds the three objects that we will display, 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*.
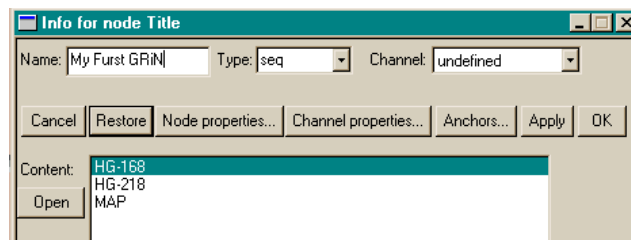
In this Info box, we see that there is no pointer to a file or URL, but a list of the objects that are contained inside this structure node.

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 document 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

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

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-218*) by clicking on it once. Using either the right mouse button, or the Tools menu, select Properties, which brings up the following window:
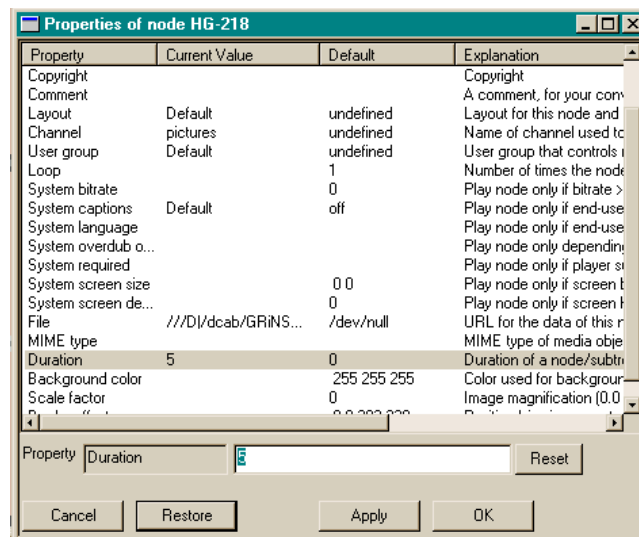
Using either the arrow keys or the mouse, select the line labelled *Duration*. In the attribute editing area at the bottom of the box, replace the value 5 with 2; this reduces the time that this object is visible from five seconds to two seconds. Confirm with OK. Now, run via Play. The first image will 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 last image (the map of Amsterdam) from the end of the presentation to the front 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:
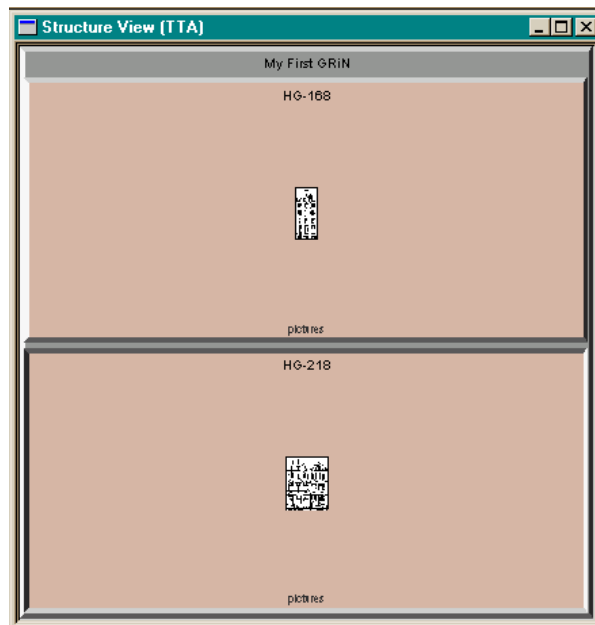
Next, select the object labelled *HG-168* by clicking on it once. Now, go to the Edit menu and select Paste node -> Before, as shown below:
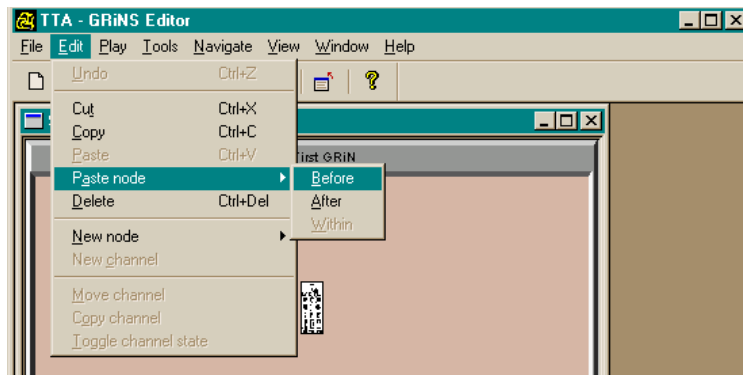
This puts the contents of the clipboard before the item you selected, resulting in:
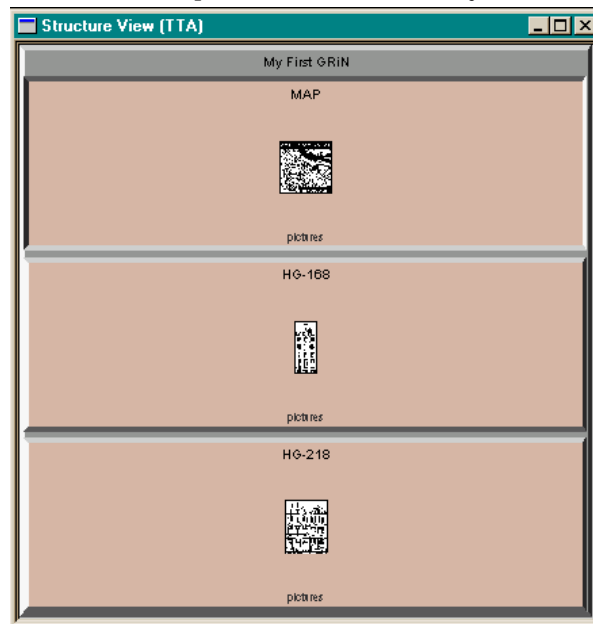
(If you had selected After, it would have placed map after this node.)

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 first object) and place it at the end of the presentation. To do this:
1. select the Map object by clicking on it once, and then selecting Copy from the Edit menu
2. select the HG-218 object once
3. paste the Map object selecting the sequence Edit->Paste node -> After from the Edit menu

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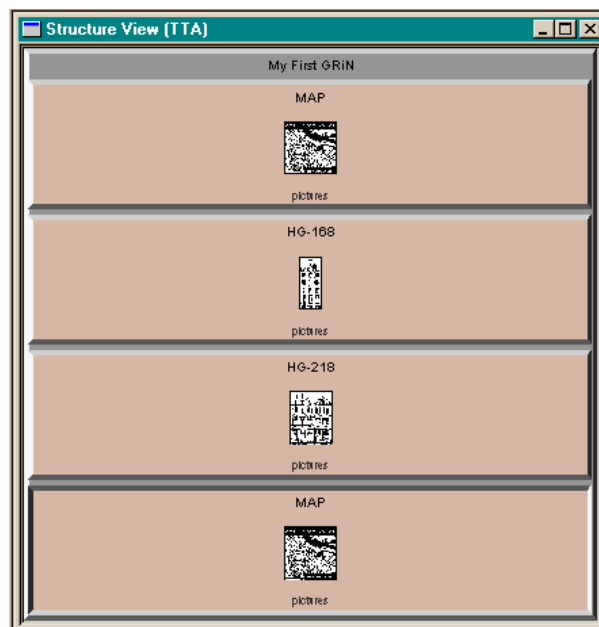
By selecting Play, you can view the work just finished.

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: Adding Parallel Objects and Audio a try when you are ready.

GRiNS/SMIL Tutorial Guide

Tutorial 2: Working with Parallel Nodes and Audio

In this tutorial we create a simple SMIL presentation that displays a sequence of image assets in three separate regions on the screen. The images will be accompanied by a looping piece of background music.

In order to create this presentation, we:
1. open the editor and select the *Extended-Slideshow* template;
2. associate media assets with each of the object references in the presentation;
3. preview the presentation;
4. fine-tune the timing relationships between objects using property sheets;
5. further manipulate timing relationships using the GRiNS Timeline View; and
6. 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.
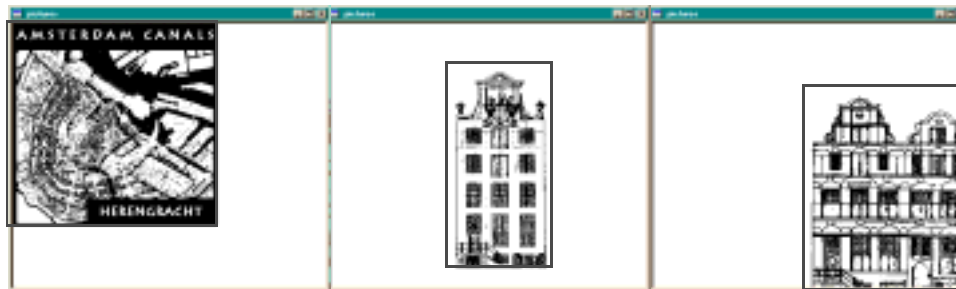
(a) use of Top-L Region   *(b) use of Mid Region*   *(c) use of Bot-R Region*

21

Open the Extended-Slideshow.SMIL template

Start-up the GRiNS Editor and open the *Extended-Slideshow.smil* template by selecting New from the File menu, and browsing the *Template* folder in the GRiNS system folder. When the document opens, the Structure View is activated. This will give you the following Editor view:
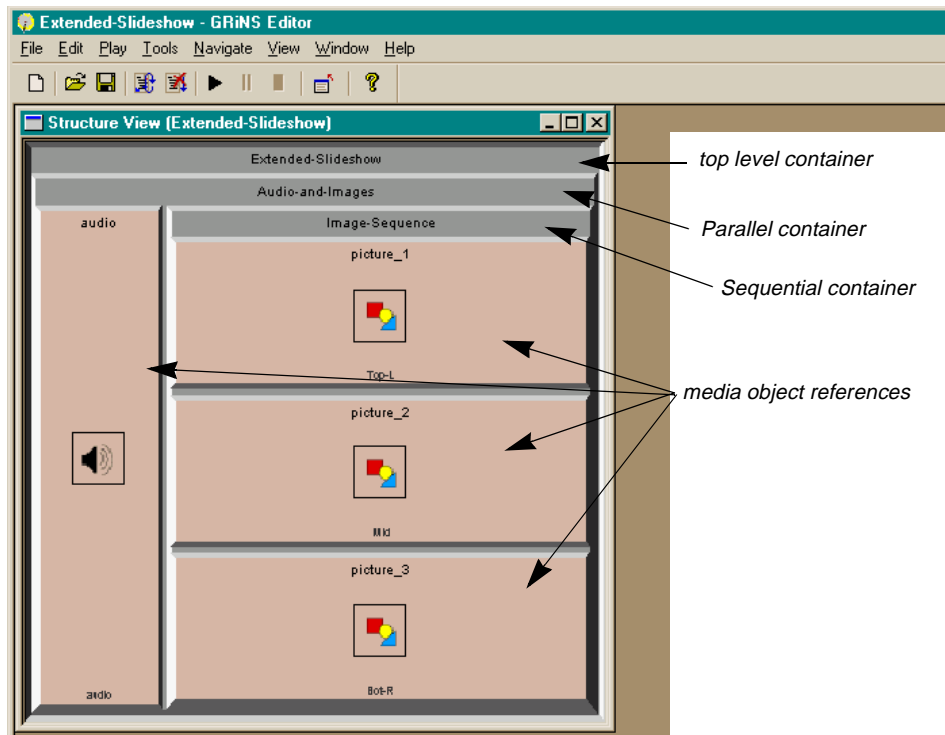
As in the first tutorial, we recommend you save a copy of this presentation in the *My GRiNS Tests* folder, this time using the name *TTB.smil* for your file.

Understanding the GRiNS Structure View

In the first tutorial, we saw that a grey structure box with a stack of three items was the GRiNS method of representing a SMIL sequential structure container (SEQ). In this document, we see the same structure in the container named Image Sequence. There is also an additional structure container: a parallel (PAR) container labeled *Audio and Images*. This container holds one structure node (the sequential container *Image Sequence*) and one data object reference node, labeled *audio*.

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 template corresponds to the following SMIL structure:

```
<seq id="Extended-Slideshow" >
    <par id="Audio and Images" >
        <audio id="audio" ... />
        <seq id="Image Sequence" >
            <img id="picture_1" ... />
            <img id="picture_2" ... />
            <img id="picture_3" ... />
        </seq>
    </par>
</seq>
```

Spend a few moments to make sure you 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!

Associating the Structure with the Data

As in Tutorial 1, the first step in the creation of a template-based GRiNS presentation is the association of the media assets available with the overall structure of the presentation. Note that by separating structure from content, we 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 we'll see in a later tutorial.

Fill in the Images for Use in the Show

Using the techniques learned in *Tutorial 1,* fill in the three images in the *Image Sequence* structure container. You can used the images in the *Data* folder provided with this tutorial, or you can use any other images that you like. For each data object, fill in the data object references for *picture_1, picture_2* and *picture_3* as follows:
1. select the object reference you wish to change (it is highlighted);
2. using the right mouse button, select Info (or use the Tools -> Info menu combination).
3. rename the object reference to something associated with that picture and set the file to point to the image you wish to use.
4. confirm these selections
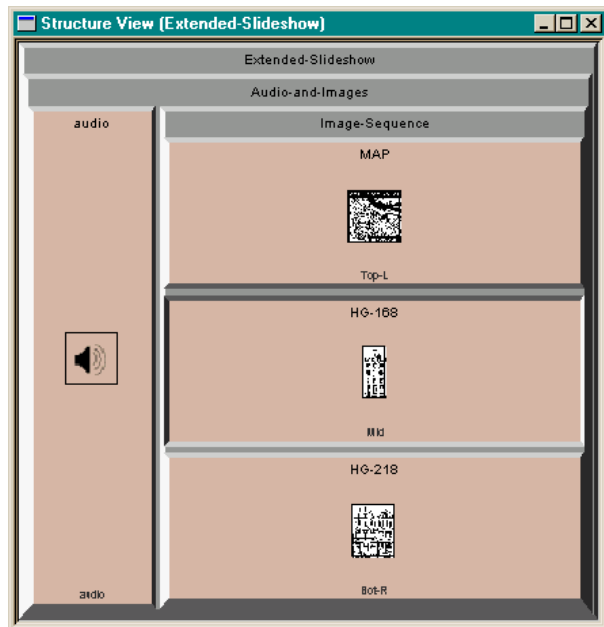
Being creatures of habit, we simply reused the names and images *map.jpg, h168.jpg* and *h218.jpg* we had in Tutorial 1. The result is the Structure View at left.

If you would like to get a taste of the presentation as it now stands, select Play from the Play menu, the VCR-like control or the right-mouse shortcuts. You will get a pop-up message saying that the Audio file has not been defined — this is not an error, but only a warning to you that the presentation is not yet complete.

Note:

1. (The GRiNS Player can render most image types used on Windows, Macintosh or UNIX platforms, but if you want to play the presentation on another SMIL compatible player, such as the RealNetworks G2, you should make sure that the data types you have selected are supported by that player.

Fill in the Audio Background for the Show

As a last step, select the data node named *audio* and assign it a media asset. We do this in the same way that all other media assets have been assigned: using the Info command from the Tools menu (or under the right mouse button).
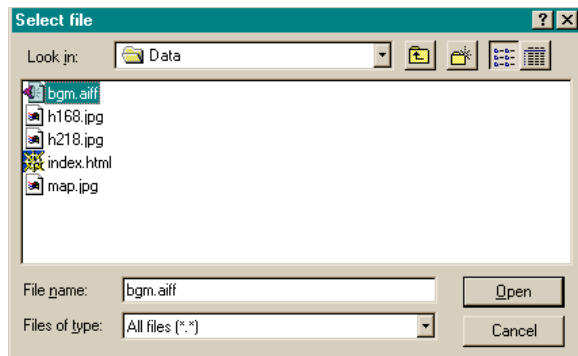
The Data folder contains a short piece of background music; it is named *bgm.aiff*. You can also use some other piece of music in any GRiNS-compatible format, but we suggest you use *bgm.aiff* because we will make use of its properties (such as its duration) later in this tutorial.

In the same way that you assigned the images to the image data nodes, assign *bgm.aiff* to the audio node now. You do this by selecting the audio node in the template and then pointing the Editor to the *bgm.aiff* file in the *Data* folder.

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 we 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 change the duration to loop, 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 the Duration of 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 Tools menu, select Properties, which brings up the properties window for the audio node:
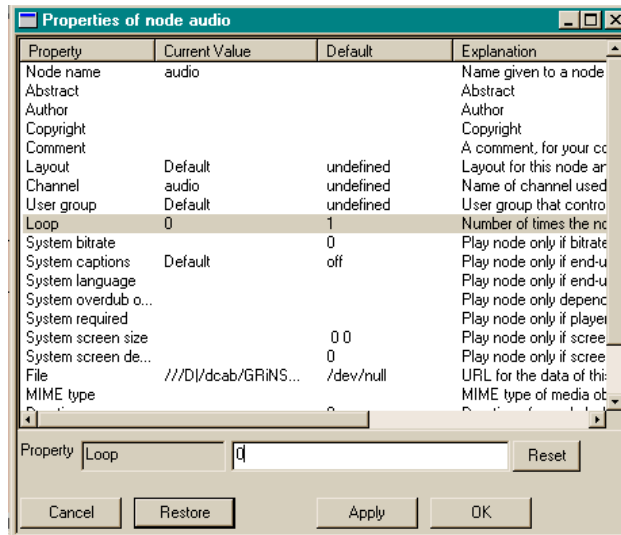
Using either the arrow keys or the mouse, select the line labeled Loop. In the attribute editing area at the bottom of the box, replace the value 1 with 0, which is the SMIL way of saying Indefinitely. Confirm with OK.

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.

There are several ways of telling GRiNS to do other things with your presentation. Just for fun we will have a look at a few things you can do with looping and duration in the next section.

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, we 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 0 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.
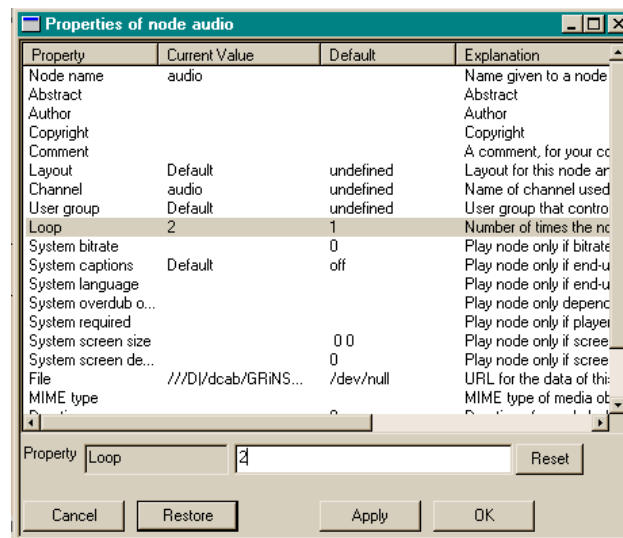
Setting an explicit duration

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 we had set the Loop attribute associated with the *bgm.aiff* file to 0. As we heard, this means that the object keeps repeating indefinitely. When we 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 0 (SMIL's way of saying indefinate)
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:
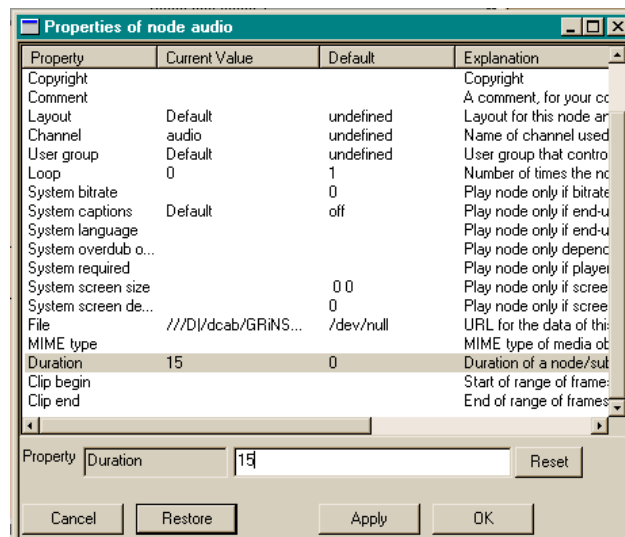
What happens: the images play in sequence, and the audio plays for the duration we have defined (15 seconds).

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 we have defined a Duration of 15 seconds for the audio and a Loop counter of 0 (for indefinate 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
Author			Author
Copyright			Copyright
Comment			A comment, for your convenience only
Layout	Default	undefined	Layout for this node and its descendan
Channel	Mid	undefined	Name of channel used to render this no
User group	Default	undefined	User group that controls rendering of th
Loop		1	Number of times the node should loop (
System bitrate		0	Play node only if bitrate >= value here
System captions	Default	off	Play node only if end-user wants captio
System language			Play node only if end-user prefers this la
System overdub or caption			Play node only depending on end user
System required			Play node only if player supports these
System screen size		0 0	Play node only if screen bigger than thi:
System screen depth		0	Play node only if screen has more bits t
File	///DI/dcab/GR...	/dev/null	URL for the data of this node
MIME type			MIME type of media object
Duration	9	0	Duration of a node/subtree (in seconds
Background color		255 255 255	Color used for background of window

Property Duration | 9 | Reset

Cancel | Restore | Apply | OK

Explicitly Selecting the End of a Parallel Group

Let's think about what we really want this presentation to do
1. we want a sequence of slides, each of which last the duration defined on the individual attribute pages for the images; and
2. we want the background music to last as long as the total time of all the images.

In this example, 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, we want to edit the properties of the parallel structure container that holds both the sequence of images and the audio node.
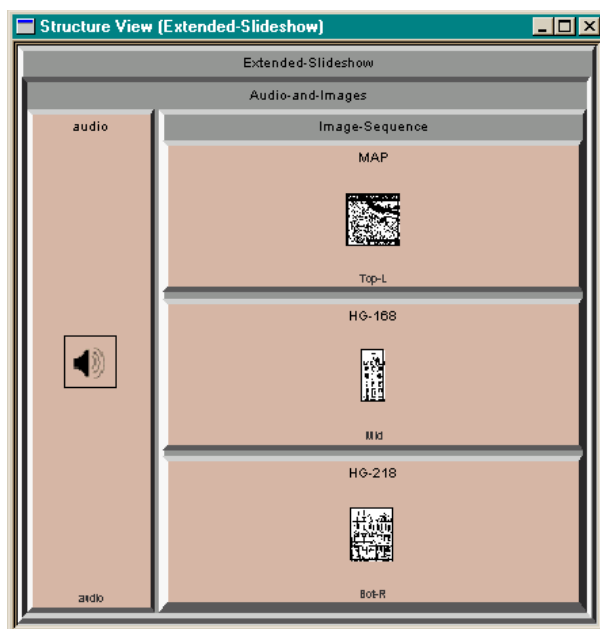
To locate this parallel container, open the Structure View. You will see the structure at left.

Select the structure node named *Audio and Images.* This container has two elements: the *audio* object and a sequential container named *Image Sequence* holding the three images.

What we want is to keep looping the audio node until the last element in the sequential node is finished. In other words, *audio* terminates when *Image Sequence* 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.

We should first change the *audio* node to have a 0 duration and 0 loop, which signifies that we are not really interested in its duration and loop behaviour.
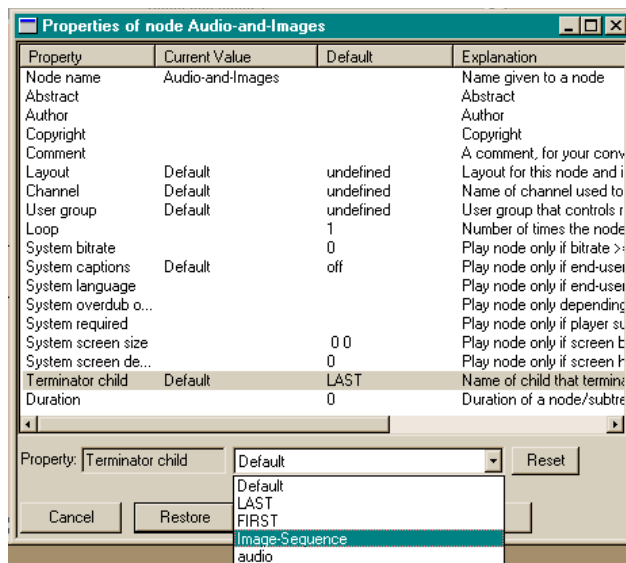
Next, to set this property, select the *Audio and Images* node and pull up its properties page using the Properties directive in the Tools menu or under the right mouse button. Using the arrow keys or the mouse, highlight the Terminator child property, and bring out the pull-down menu in the attribute editing area at the lower side of the property page. The sheet before the change is accepted is shown.

We set the Terminator child to the structure node named *Image-Sequence.* (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 *Image-Sequence* 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 accordingly.

You should try other combinations as well. Note that if the Loop value of any node is set to indefinate, 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.

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:
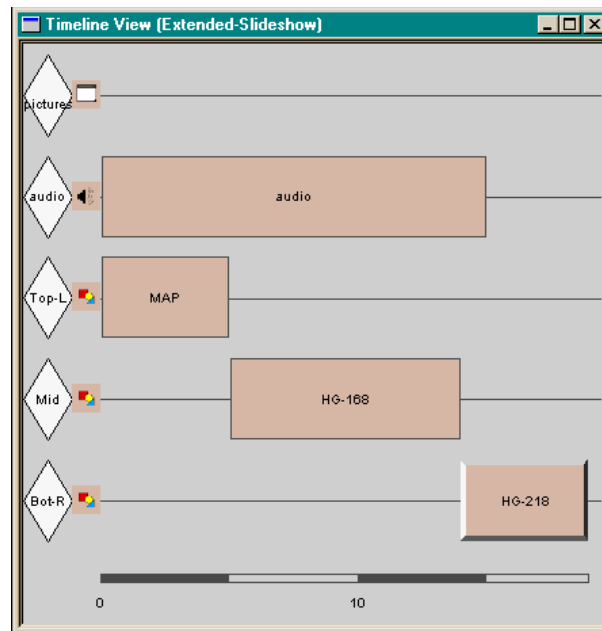
This view shows four objects: the *audio* object and *picutre_1*, *picture_2*, and *picture_3*. If we select *picture_2* by clicking on it once, we can bring up its properties via the right mouse button. Find the *Duration* property and change it from 5 to 2 seconds, as shown below.

The property sheet for this node contains all of the SMIL and GRiNS-related properties associated with this node. The use of most of these properties will be
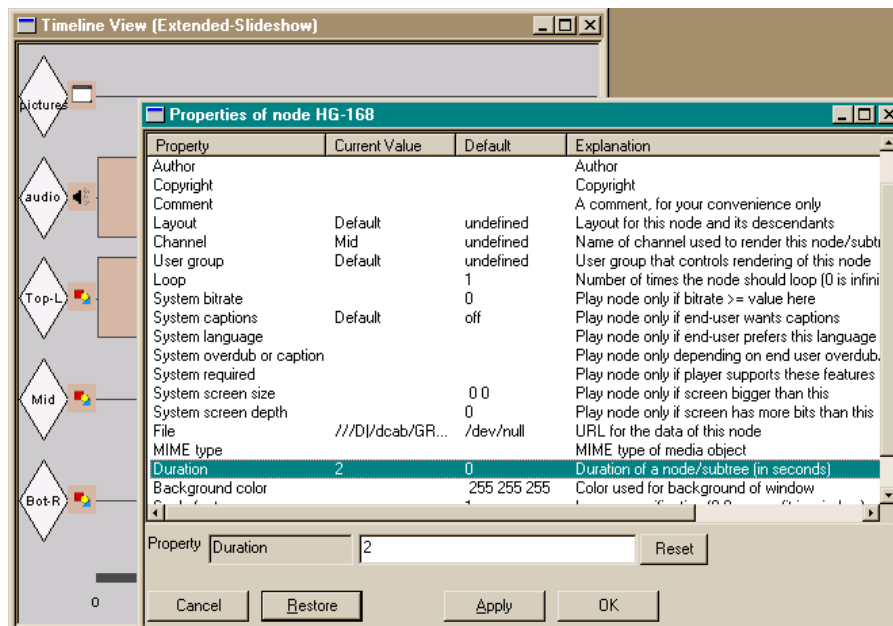
covered in other tutorials. You can also consult the GRiNS User Manual and the GRiNS Reference Manual, both of which are available from the GRiNS Web site.

The result of this is that the length of the *picture_2* object is now scaled to two seconds automatically. The runtime behavior can be confirmed by previewing the presentation.

Fine-Tuning via Synchronization Arcs (Sync Arcs)

Let's suppose that we wanted to take our presentation-under-construction and make some fine-grained adjustments to timing. For example, let us assume that we want the audio fragment to begin slightly later than the first image. Perhaps we are doing this to balance the delivery load of the presentation, or maybe we 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. If you select the first picture (picture_1),
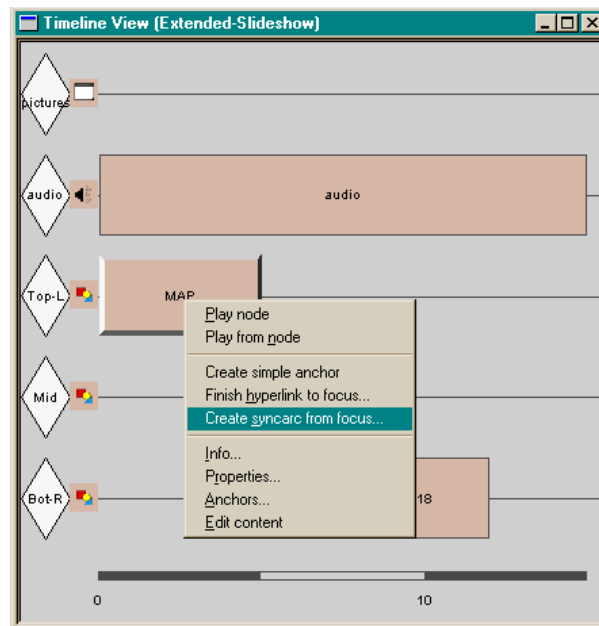
This will define the source part of a sync arc. The source is colored amber to highlight it in the Timeline View.
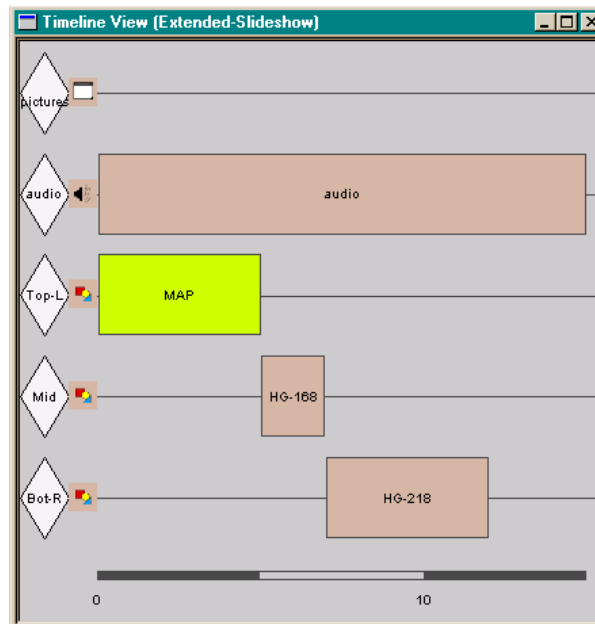
Once the amber node is shown, you should select another node in the timeline to identify the target of the sync arc. In this tutorial, click once on the node named Audio. This will bring up the Sync arc properties box, as shown top right. The GRiNS default is to define a relationship between the end of the source to the beginning of the target. In our case, we change *From* to *Begin* and give a delay of 2.5 seconds.

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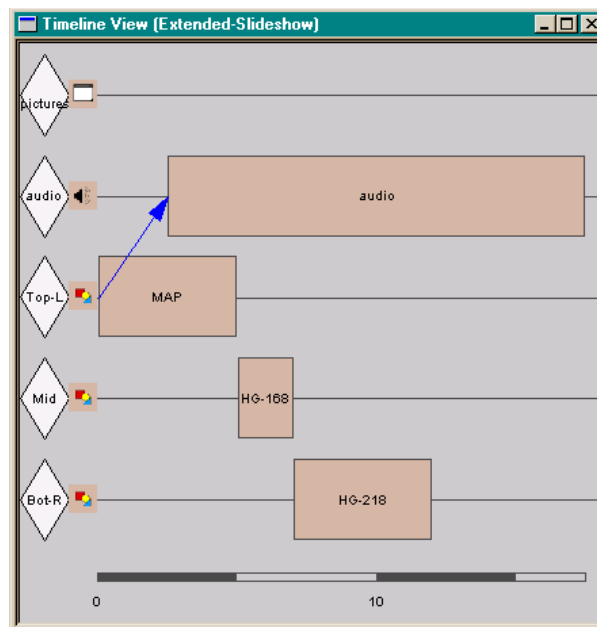
Once a sync arc is created, you can edit its properties by selecting the arrow head associated with the arc. If you make changes, the timeline will be redrawn automatically for you.

Notes:
1. In a future version of GRiNS, you will be able to select synchronization relationships at specific marker points in an objects rather than only at the beginning or the end of an object.
2. SMIL places structural restrictions on the scope of sync arcs. In SMIL V1.0, both the source and target must be have a sibling relationship. This is expected to be relaxed in version 2.0.

Better Understanding Timing in Parallel and Sequential Structure Nodes
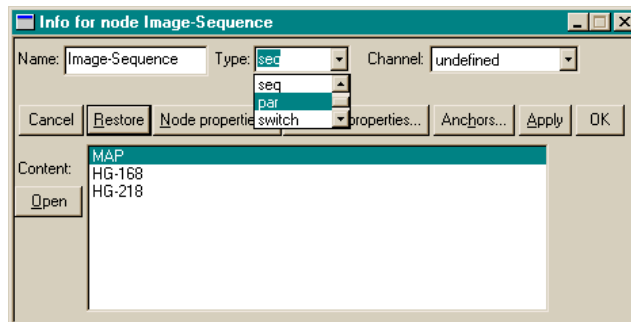
Now, go to the Structure View and select the Image Sequence structure node and bring up the Info box. In the middle part of the top of this box is the Type selection box. If you highlight the alternatives, you get the image at left.

Select *par*, which will take the image sequence and turn it into a parallel structure group in which all of the images are displayed simultaneously.
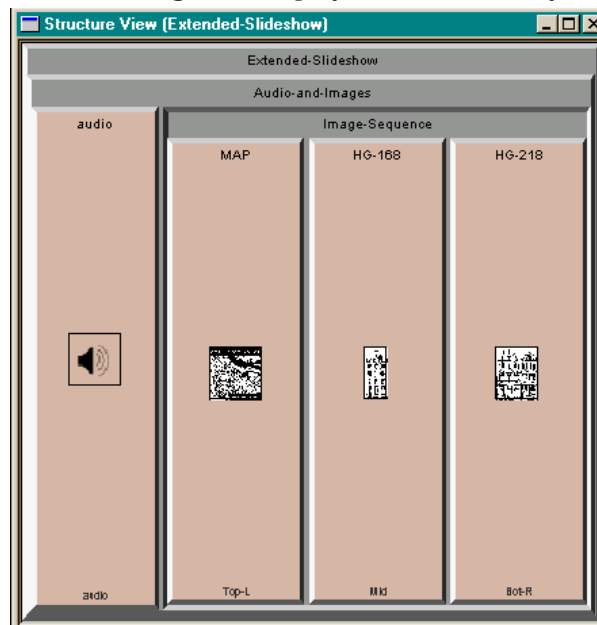
If you confirm the selection of par and then preview the presentation using Play from the menu bar, the Play icon or from under the right mouse button, you will see that the images are played together. *Picture_1* and *picture_3* are displayed for five seconds each, and *picture_2* for three seconds. The audio lasts as long as the longest image (instead of the sum of the image times).
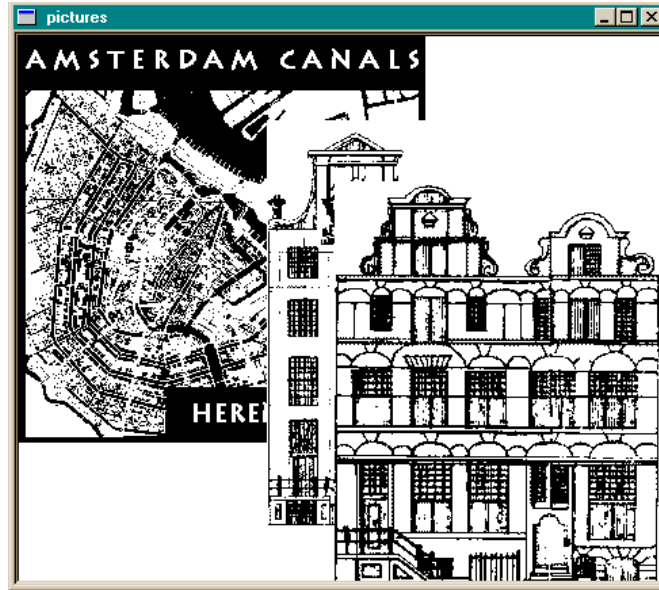
The presentation as it now stands has a delay defined for the start of the audio. This means that the Map image is shown for 5 seconds, by the music starts 2.5 seconds after the Map has been shown. This behavior can best be seen by looking at the timeline view. Here we see the images all starting in 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 we won't be needing it in this example.
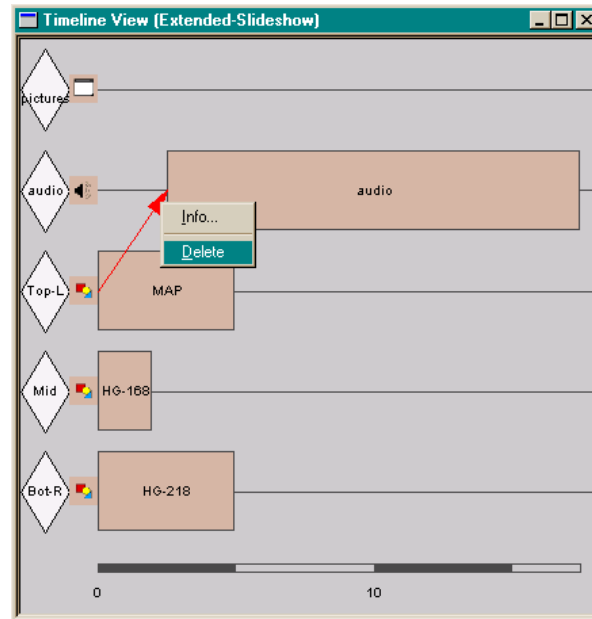
Selecting Layout Regions

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

In the next tutorial, we take a deeper look at Resource Channels. To get a taste of what this kind of presentation control GRiNS gives you, we 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 *Image Sequence* — this is the *MAP* object. Next, bring up the Info box for this image by selecting Info from

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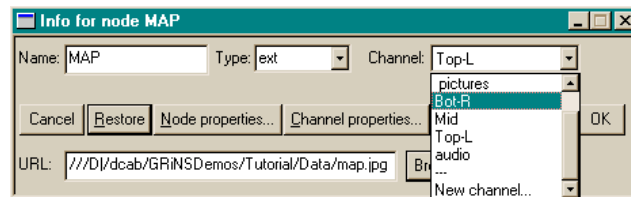
If you select Bot-R, the *MAP* image will be displayed at the bottom-right part of the screen. Select Bot-R, confirm the selection and then preview the presentation using PLAY. If you also change the second image (*HG-168*) to display its output in Bot-R, all three images will be displayed in the bottom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

GRiNS/SMIL Tutorial Guide

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

The purpose of this tutorial is to create a the *Extended-Slideshow.smil* SMIL template that was used in Tutorial 2. The *Extended-Slideshow.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 fill a template with media objects, 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 template for later use.

Since we are building a template rather than a full presentation, we will not make extensive use of presentation preview in this example.

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 Extended Slideshow

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 Extended Slideshow 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 conventional HTML browsers, 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.

Notes:

1. Use of SMIL Basic Layout is optional in SMIL-V1.0, but all SMIL-compliant players are required to support SMIL Basic Layout. This provides you with a measure of transportability across SMIL players.

2. Unfortunately, only limited facilities exist in version 1.0 of SMIL for doing lots of transitions and other graphical effects on a presentation, but these facilities can be expected in SMIL 2.0. Future versions of the GRiNS Editor for SMIL will preview some of these functions as a part of helping the SMIL standard evolve.

Open the GRiNS Editor and select the EMPTY.SMIL template
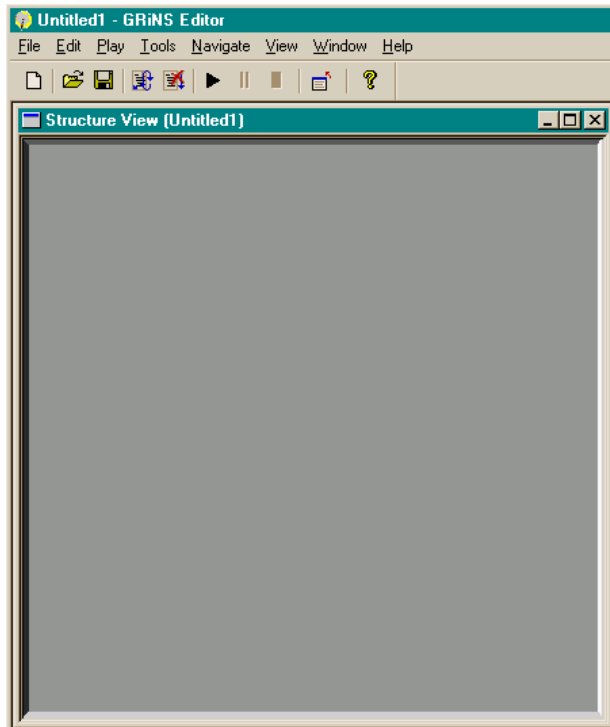
If you have not already done so, start-up the GRiNS Editor; go to the File Menu and select New.
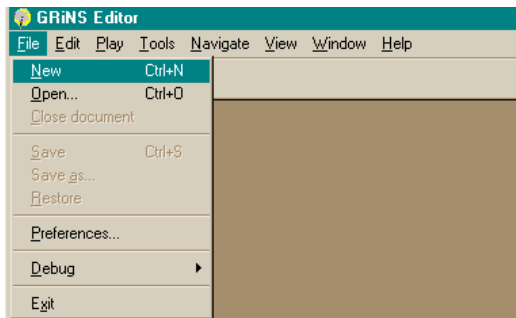
You will be asked to select a template file, as shown below:
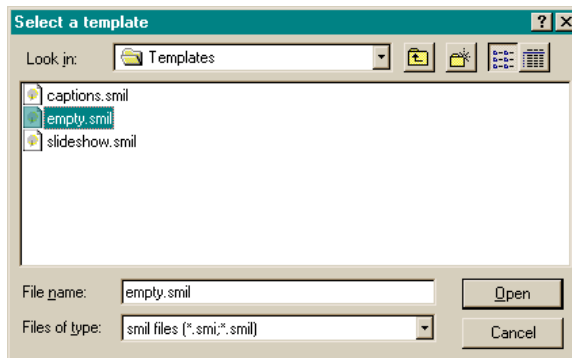
Select the template named *Empty.smil*. The Empty.smil template contains the SMIL boiler-plate information required to build a SMIL-based presentation.

When you open the *Empty.smil* document, you get the default GRiNS Structure View, which in the empty document contains a single empty 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.
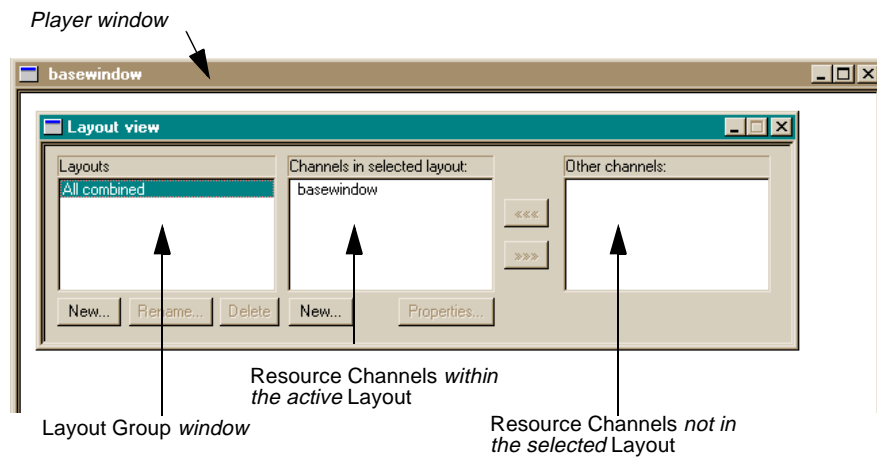
NOTES:
1. Template files are a clue to the Editor on the format you wish to ultimately export.
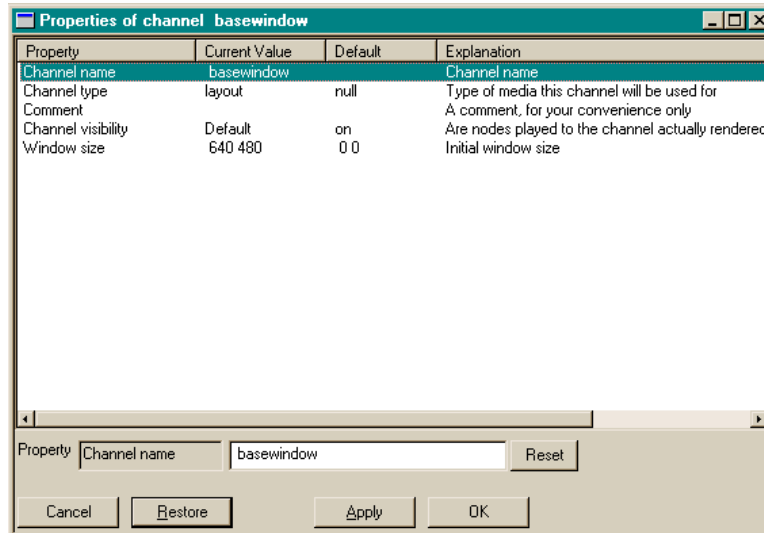
2. If you have a PRO version of the GRiNS Editor, you can create SMIL, CMIF, ASX (Netshow) and Quicktime documents, plus documents optimized for particular platforms and streaming media formats. In the general-purpose version of GRiNS, you can create either SMIL or CMIF presentations.
3. Some implementations of GRiNS allow you to convert presentations from one format to another. The final format of the presentation is determined when you save the file (with the SAVE AS option).

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:

Player window

Resource Channels *within
the active* Layout

Layout Group *window*

Resource Channels *not in
the selected* Layout

GRiNS provides you with a *basewindow*, 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 *basewindow* by selecting the name *basewindow* in the middle (Channel) window of the Layout View, and then selecting

Properties of channel basewindow

Property	Current Value	Default	Explanation
Channel name	basewindow		Channel name
Channel type	layout	null	Type of media this channel will be used for
Comment			A comment, for your convenience only
Channel visibility	Default	on	Are nodes played to the channel actually rendered
Window size	640 480	0 0	Initial window size

Property: Channel name basewindow Reset

Cancel   Restore   Apply   OK

the Properties button. This brings up the following Property Sheet for the *basewindow*:

Define SMIL Regions via GRiNS Resource Channels

A SMIL presentation allows you to put any number of regions inside a presentation base window. We start by moving the mouse to the section labeled Channels, and then selecting the New... button. This will bring up a dialog box that allows you to define the Channel.

New channel

Channel name: Top-L

Choose type: html
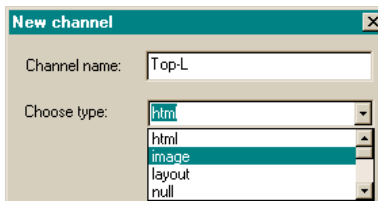
html
image
layout

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 selecting it in the Channels list and pressing the Properties button. The property sheet contains the following elements:
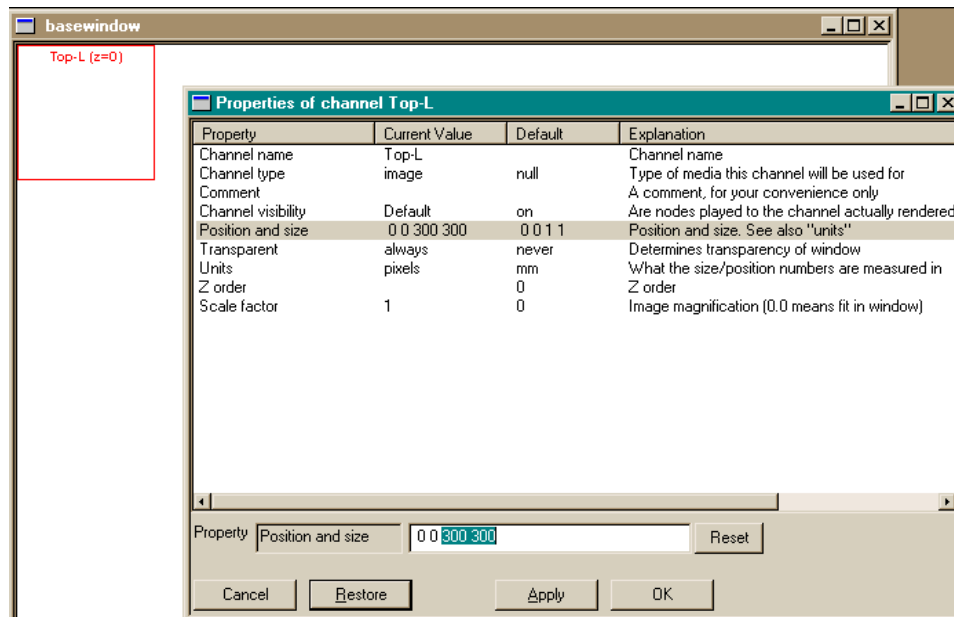
(The default window is shown at top left, and the new window parameters are shown in the property sheet.) You can change the location of the window and its size by entering the parameters directly into the property sheet. This is useful if you want pixel-level control over your application.

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, move your mouse to the a position within the Channel's box and select *Resize* from under the right mouse button. This will allow you to both move and reshape the Channel, with the positioning and pixel sizes of the Channel displayed in the middle of the Channel's space.

The following illustration shows the view that you get:
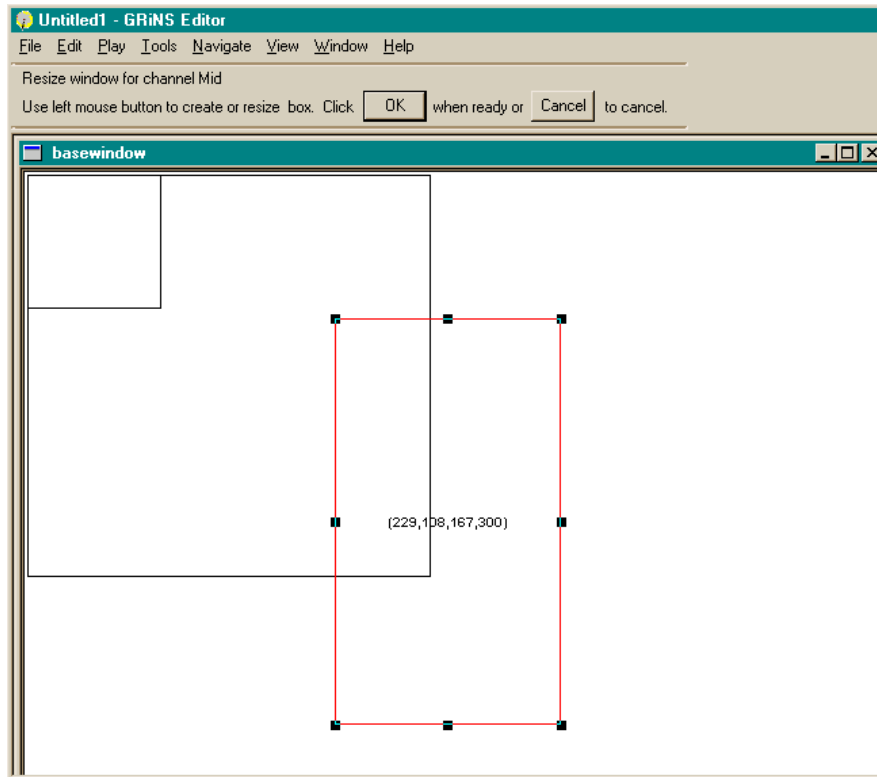
Once you are done with the positioning, you can accept the changes by clicking OK at the top of the window.

GRiNS/SMIL Tutorial Guide

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:
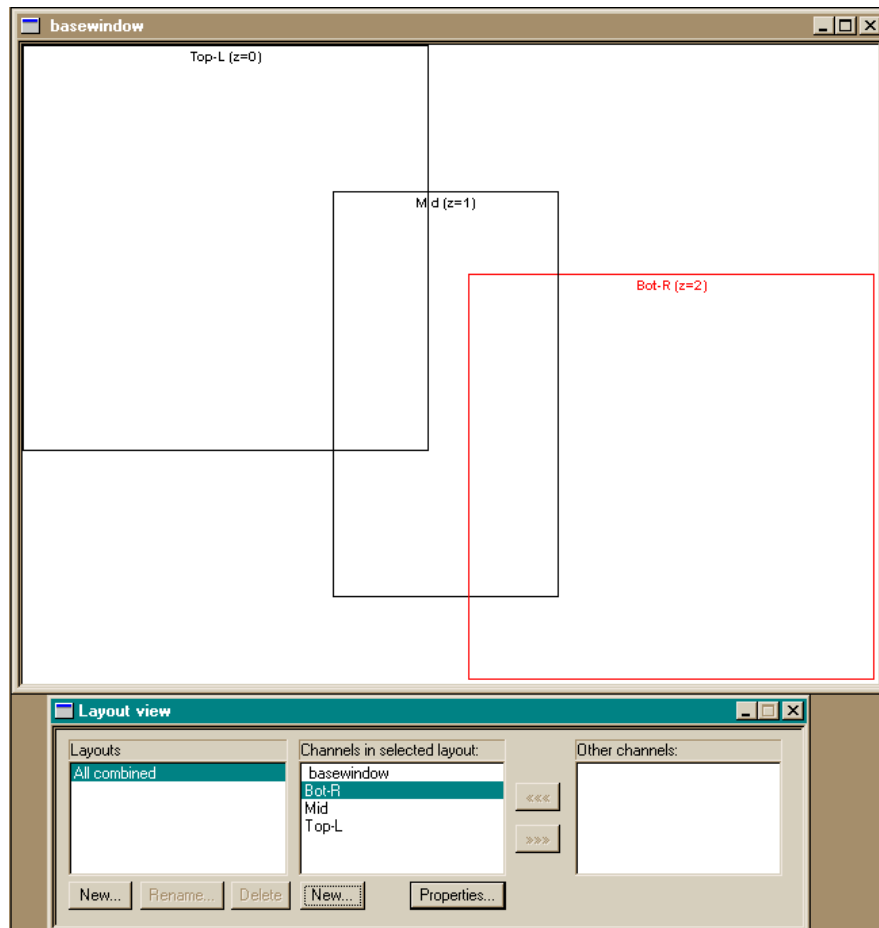
At this point the presentation contains one base window and three rendering
Channels, named *Top-L*, *Mid*, and *Bot-R*. Each of the Channels is set to render
images in the format defined by the author. Note that we don't know if the images
will appear in parallel or sequentially, or if each Channel will be used for one or

more images. This depends on the specification of the presentation in the Structure and Timeline views.

Defining the Structure and Contents of the Presentation

The *Extended Slideshow* defined 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
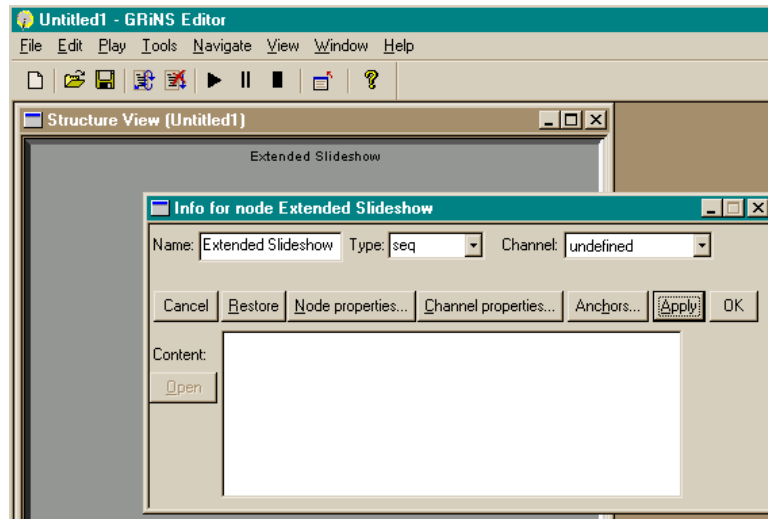
structure node *Extended Slideshow*. The other properties are already filled in by the Editor. (The illustration shows the name placed in the node after Apply was selected.)

Note:

55

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.
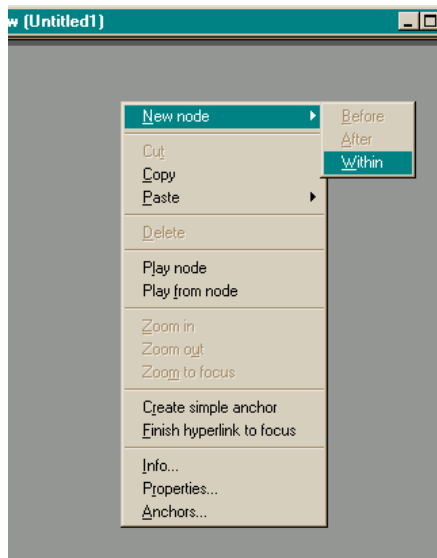
We start the development of the presentation by creating a new node within the *Extended Slideshow* container by selecting New -> Within from under the right mouse button.

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.

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), set the URL to *put-audio-here.aiff* and select *New Channel* under the channel window. Setting the URL to a name ending in *.aiff* (or another extension that is used for audio files, such as *.wav*) is necessary: the New Channel command sets the type of the newly created channel based on the media type you are going to use it for.
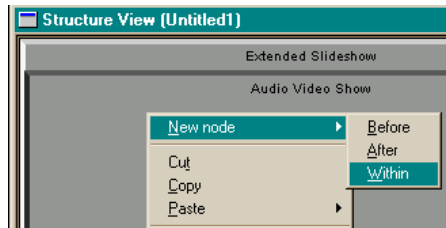
When you select New Channel, you will be given a *Channel name* pop-up box. You should name the new channel *Audio*. This resulting Info view is:
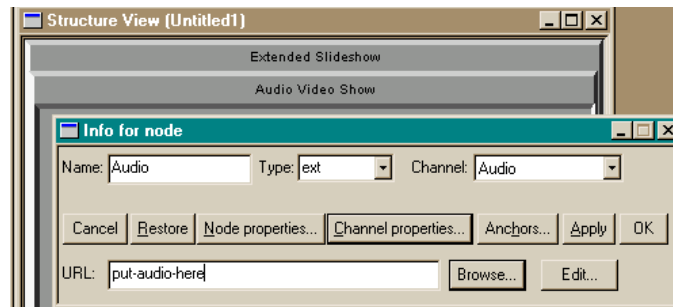
After creating the audio node, the Structure view will look as follows:
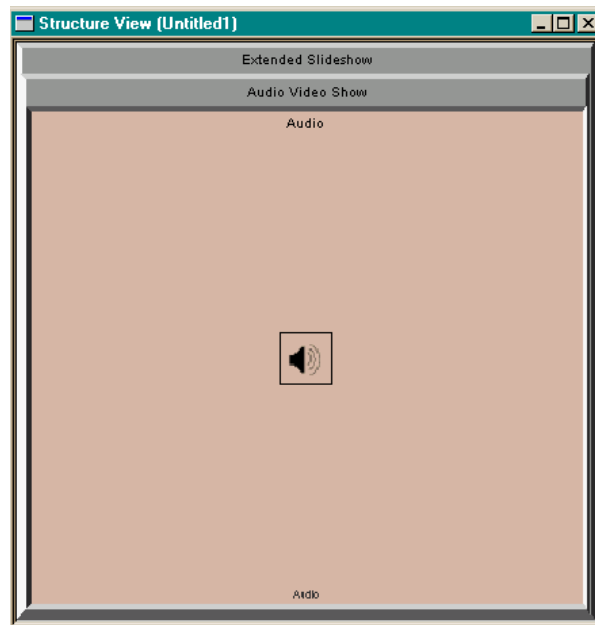
What we have defined so far is a presentation structure with a single audio media reference. We now want to define the sequence of images in parallel with this object.
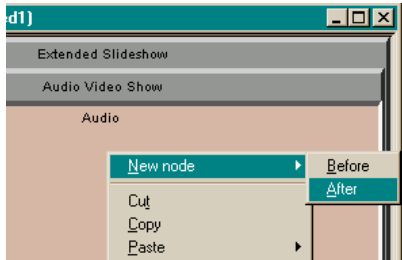
First, we need to create a sequential structure container. There are several ways of doing this, the most obvious of which is to select New Node > After from the right mouse button. We will create a sequential structure node after the Audio object.

Once the new node is created, open that node's Info box and define the high-level characteristics of the node. Name it Image Sequence, give it a type of *seq* (for a sequential node) and leave the channel type undefined. The resulting Info box is:
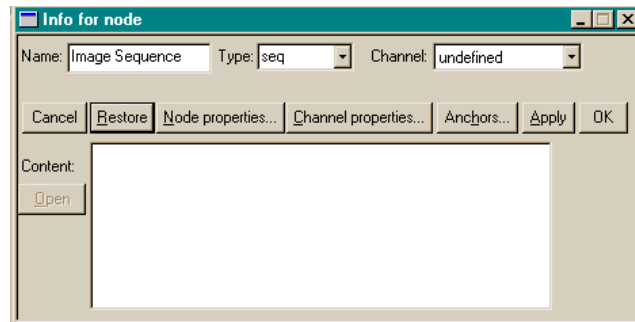
When you accept the changes, the Structure View is:
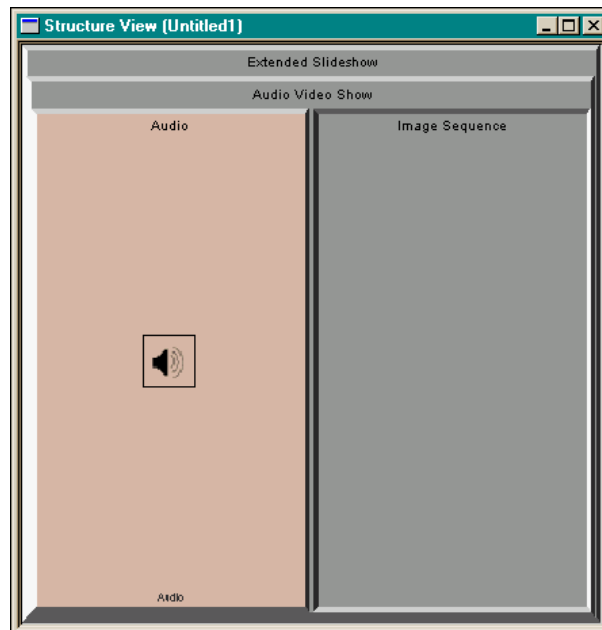
All that is left to do is to fill the sequential node with the image references. We start by defining a new node within the Image Sequence container:
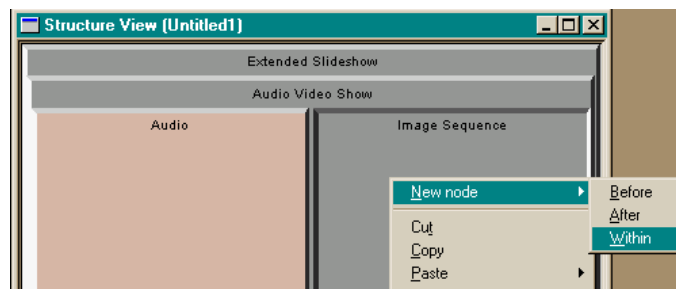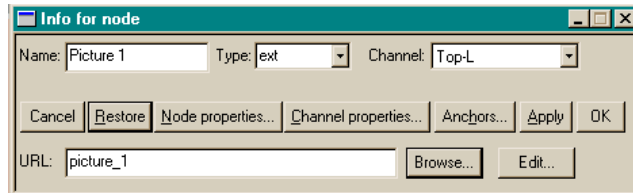
We open the Info box associated with this node, and define an image reference. The name is *Picture 1*, the type is *ext* (for external) and the channel that we will use to display this image is *Top-L*, which is one of the channels we defined earlier in this tutorial.

The URL 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). The update Structure view is shown on the next page.

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
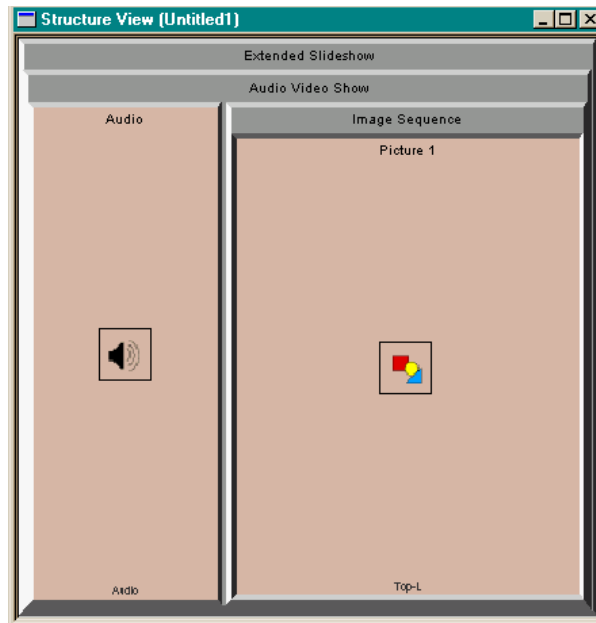
copy and paste facility by selecting copy from Edit menu and then Paste After. If you use copy and paste, be sure to open the Info boxes for each node and change the node name, the channel assignment (Mid for *Picture 2* and Bot-R for *Picture 3*), and the name of the file/URL to be referenced.
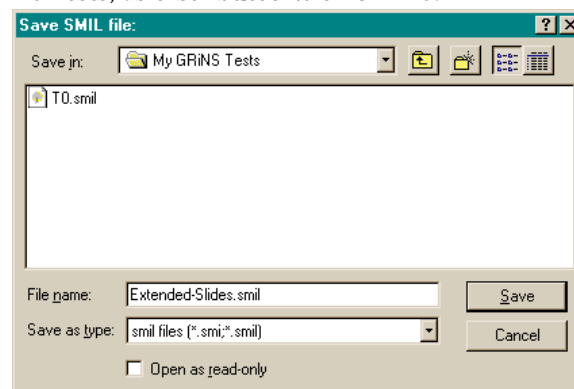
The resulting Structure Views after each step are:

The template is now complete.

Saving the Presentation

After creating the template, we need to save it somewhere. While this would normally be in a Templates directory, it is probably best to write this file in the folder My GRiNS Tests, as distributed with GRiNS.

If you wish, you may tag the file as being read-only.

Defining the durations

You have now defined a presentation template that contains three images which get projected into three separate SMIL regions and one audio file. Since images have no inherent duration, they will each be displayed for the shortest time increment support by the SMIL player you use. Even on slow players, this is not very long. If our presentation consisted of an audio object that was played in parallel with a single image, that image would usually be visible for the duration of the audio object. In our presentation, we have three images played in sequence with one audio node, so the player will have no way of knowing how much time to allocate to each image.

We could leave the problem of relative durations up to the presentation author, but it is often helpful if we define default durations for images (and text and other non-continuous media) at the time we define the template. This will make the presentation a bit easier to create and preview, especially if the user wants to make use of the of the play node facility in GRiNS, which allows a sub-part of the presentation under construction to be previewed.

Setting durations in nodes is something that was covered in earlier tutorials. The basic of the process are:
1. Select the node for which you want to set a duration.
2. Go to the Tools menu and select Properties (or select Properties from under the right mouse button).
3. A dialogue box will appear containing all of the properties for this node. Scroll down using the arrow keys until Duration is highlighted. In the edit area below, type 5 (for 5 seconds).

You should set a standard five second duration for *Picture 1*, *Picture 2* and *Picture 3*.

Previewing the Presentation

The Extended Slideshow template contains all of the structure and object information required to construct the presentation. Unfortunately, as it now

stands, each of the objects only contains a place-holder name. As a result, there isn't much to see in this presentation. (If you used actual file names or URL references, then the presentation may contain actual data references.)

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.
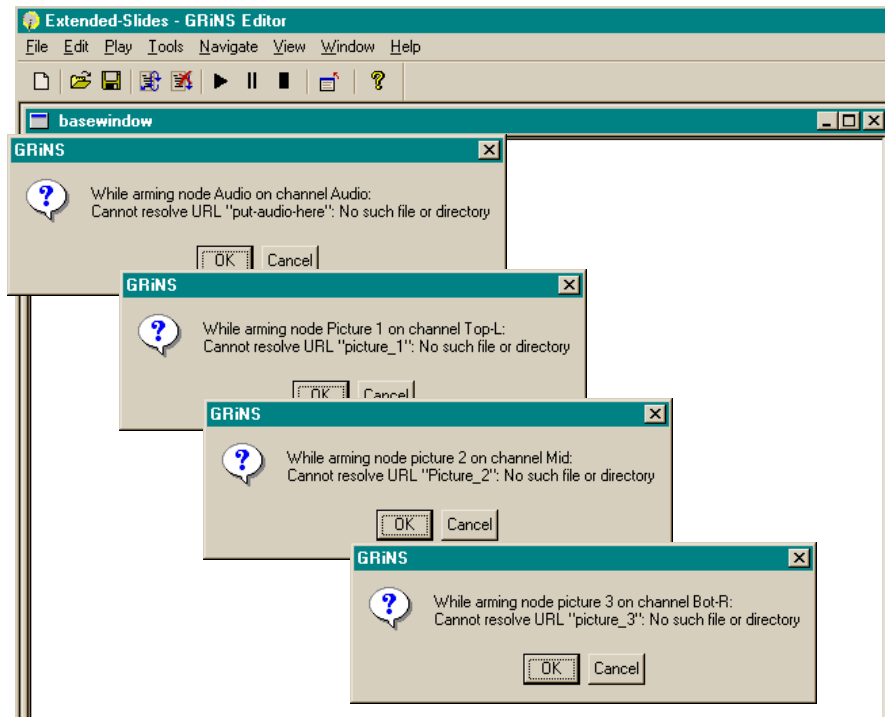
Tutorial 4: Working With Anchors, Links and Hypermedia

This tutorial is being revised and will be available shortly.

GRiNS⁄SMIL Tutorial Guide

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 ⁄ SMIL Tutorial Guide

GR*i*NS Quick Reference Information

SMIL Compliance Information

The GR*i*NS Editor for SMIL and the GR*i*NS 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 GR*i*NS 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)
image/cgm		no	no	no	no

Table 1:

MIME type	Extensions	Windows 95/98/NT	Mac	SGI Irix-6.3	Sun Solaris-2.5
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-anymap	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
	pm	no	no	no	no
	ras	yes	no	no	no
	tga	yes	no	no	no

Table 1:

MIME type	Extensions	Windows 95/98/NT	Mac	SGI Irix-6.3	Sun Solaris-2.5
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	no	yes	no
RealVideo	rv, rm	yes	no	yes	no
RealText	rt	yes	no	yes	no
RealPix	rp	yes	no	yes	no

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*).