

# ***NeroVision API***

## ***v2.1.2.0***

**The *NeroVision API* will only work with  
fully installed *Nero* and *NeroVision Express* versions!**

# 1. Contents

<b>1. Contents .....</b>	<b>2</b>
<b>2. License Agreement.....</b>	<b>5</b>
<b>3. Introduction.....</b>	<b>6</b>
3.1. Overview .....	6
3.2. Requirements.....	6
3.3. The NeroSDK Forum .....	6
<b>4. Quick Start.....</b>	<b>7</b>
4.1. Running the Precompiled NeroVisionAPIExample Application .....	7
4.2. Compiling the NeroVisionAPIExample.....	9
4.3. Accessing the <i>NeroVision API</i> in Your Applications .....	9
4.4. Points of Consideration .....	10
<b>5. The NeroVisionAPIExample Application .....</b>	<b>11</b>
5.1. CAboutDlg.....	11
5.2. CEstimateDoneDlg.....	11
5.3. CEstimateDlg .....	11
5.4. CNVAPIExampleApp .....	11
5.5. CNVAPIExampleDlg .....	12
5.6. CNVAPIExamplePage .....	12
5.7. CPage .....	12
5.8. CPageData.....	12
5.9. CPageFirst .....	12
5.10. CPageSecond .....	13
5.11. CPageThird .....	13
5.12. CRecorderCombobox .....	13
5.13. CSheet .....	14
5.14. CSpeedComboBox .....	14
<b>6. An Example Project .....</b>	<b>15</b>
6.1. The Project Components .....	15
6.2. The XML File.....	16
6.3. The Slide Show .....	17
6.4. The Video Title .....	17
6.5. The Menu .....	18
<b>7. Interfaces and Types .....</b>	<b>19</b>
7.1. IError Interface .....	19
7.1.1. ErrCode Enum.....	19
7.1.2. ErrCode Property.....	19
7.1.3. ErrText Property .....	19
7.1.4. XMLID Property .....	19
7.2. INeroBurnContext Interface .....	19
7.2.1. NERO_CD_FORMAT.....	20
7.2.2. NERO_WRITE_CD .....	20
7.3. IProgressCallback Interface.....	20
7.3.1. ProgressAction Enum.....	20
7.3.2. OnProgress Event .....	20
7.3.3. ShouldCancel Event.....	20
7.4. IProject Interface .....	20

7.4.1.	SetXMLString .....	21
7.4.2.	SetXMLFile .....	21
7.4.3.	EstimateDiskSize.....	21
7.4.4.	EstimateCreateNeroBurnContextTime.....	21
7.4.5.	CreateNeroBurnContext.....	21
7.4.6.	TempDirectory .....	21
7.4.7.	LastError .....	21
7.5.	IMediaInfo Interface .....	21
7.5.1.	FileType Enum .....	22
7.5.2.	FileType Property .....	22
7.5.3.	Duration Property .....	22
7.5.4.	HasAudio Property .....	22
7.5.5.	HasVideo Property .....	22
7.5.6.	Width Property .....	22
7.5.7.	Height Property.....	22
7.5.8.	AspectXByY Property .....	22
7.5.9.	FrameRate Property .....	22
7.5.10.	GetThumbnail Method.....	23
7.5.11.	GetImage Method.....	23
7.6.	IFileInfo Interface.....	23
7.6.1.	FilePath Property.....	23
7.6.2.	NumSegments Property .....	23
7.6.3.	GetSegmentMediaInfo Method .....	23
7.7.	IMediaAnalyzer Interface .....	23
7.7.1.	AddFile Method .....	23
7.7.2.	AnalyzeFiles Method .....	23
7.7.3.	LastError Property .....	24
7.8.	Project Object.....	24
7.9.	MediaAnalyzer Object.....	24
<b>8.</b>	<b>The XML Schema Definition .....</b>	<b>25</b>
8.1.	element nerovision-project.....	25
8.2.	element nerovision-project/vcd .....	26
8.3.	element nerovision-project/svcd .....	26
8.4.	element nerovision-project/dvd .....	27
8.5.	complexType ProjectBaseType .....	27
8.6.	element ProjectBaseType/label .....	30
8.7.	element ProjectBaseType/content .....	30
8.8.	element ProjectBaseType/content/videotitle.....	32
8.9.	element ProjectBaseType/content/videotitle/video .....	33
8.10.	element ProjectBaseType/content/videotitle/chapters .....	33
8.11.	element ProjectBaseType/content/videotitle/chapters/chapter-mark .....	34
8.12.	element ProjectBaseType/content/slideshow .....	34
8.13.	element ProjectBaseType/content/slideshow/image .....	35
8.14.	element ProjectBaseType/content/slideshow/image/header .....	36
8.15.	element ProjectBaseType/content/slideshow/image/footer .....	36
8.16.	element chapter-menu .....	37
8.17.	element chapter-menu/template .....	37
8.18.	element chapter-menu/thumbnail .....	38

8.19.	element transition .....	38
8.20.	complexType MenuTemplateBaseType .....	39
8.21.	element ProjectBaseType/menu .....	39
8.22.	element ProjectBaseType/menu/default-template .....	41
8.23.	element ProjectBaseType/menu/main-menu .....	41
8.24.	element ProjectBaseType/menu/main-menu/template .....	42
8.25.	element ProjectBaseType/menu/main-menu/thumbnail .....	42
8.26.	complexType ThumbnailBaseType .....	42
8.27.	simpleType AVTime .....	43
8.28.	simpleType FileLocation .....	43
8.29.	simpleType RGBColor .....	43
8.30.	simpleType TransitionType .....	44
8.31.	simpleType VideoEncoding .....	44
8.32.	attributeGroup FontAttrGrp .....	45
<b>9.</b>	<b>Known Limitations .....</b>	<b>46</b>

## **2. License Agreement**

IMPORTANT: PLEASE READ THE SOFTWARE LICENSE AGREEMENT ("LICENSE") CAREFULLY BEFORE USING THE SOFTWARE.

USING THE SOFTWARE INDICATES YOUR ACKNOWLEDGMENT THAT YOU HAVE READ THE LICENSE AND AGREE TO ITS TERMS.

The license agreement is contained in a text file, "NeroSDK\_License.txt", to be found in the root folder of the installation package.

## 3. Introduction

### 3.1. Overview

This paper, the documentation of the *NeroVision API*, contains some practical guidelines on how to use the *NeroVision Express* API (Application Programming Interface) with the help of *NeroSDK* (Nero Software Development Kit).

The SDK is available for OEMs (Original Equipment Manufacturers) and registered users of *Nero*.

The *NeroVision API* enables you to use a number *NeroVision Express 2* features in your own applications:

- Creation of slide shows
- Creation of menus
- Creation of video tracks

Supported formats:

- Video CD
- Super Video CD
- DVD

### 3.2. Requirements

This documentation assumes that *Nero* 6.3.0.0 or later and *NeroVision Express* 2.1.2.0 or later are already installed on your computer.

### 3.3. The NeroSDK Forum

We provide a forum for all users of the *NeroSDK* to get in dialog with each other at [http://www.nero.com/link.php?topic\\_id=17](http://www.nero.com/link.php?topic_id=17). We will also monitor the messages from time to time and try to help where possible.

## 4. Quick Start

The main effort in working with the *NeroVision API* is the creation of an XML file which contains information about your VCD, SVCD or DVD project, while the use of the functional API itself is pretty straight-forward. The XML file is required to adhere to an XML Schema as documented later in this paper.

Once you created your project in XML format it can be passed to the *NeroVision API* for processing.

The API estimates processing time and provides feedback on current phase and progress.

### 4.1. Running the Precompiled NeroVisionAPIExample Application

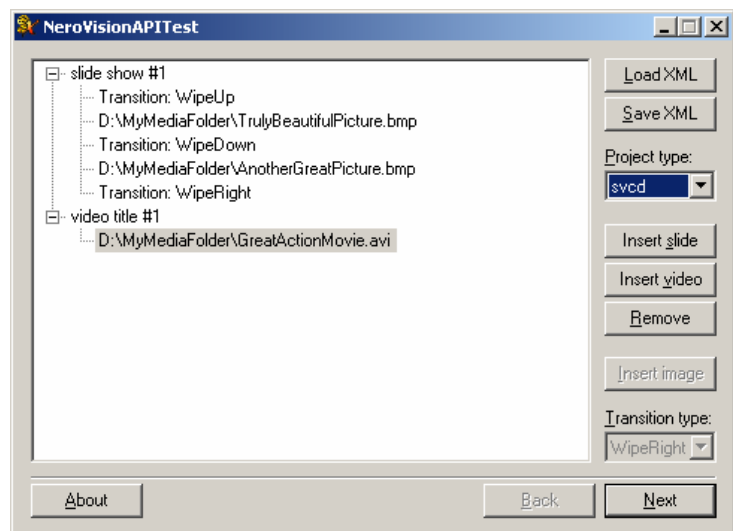
We advise you to use the example application provided with the API to familiarize yourself with the use.

To run the example, it is required that MS-XML 3.0 or better is installed.

The example is a Wizard-style dialog application that leads the user through project creation and API use with a number of tabs.

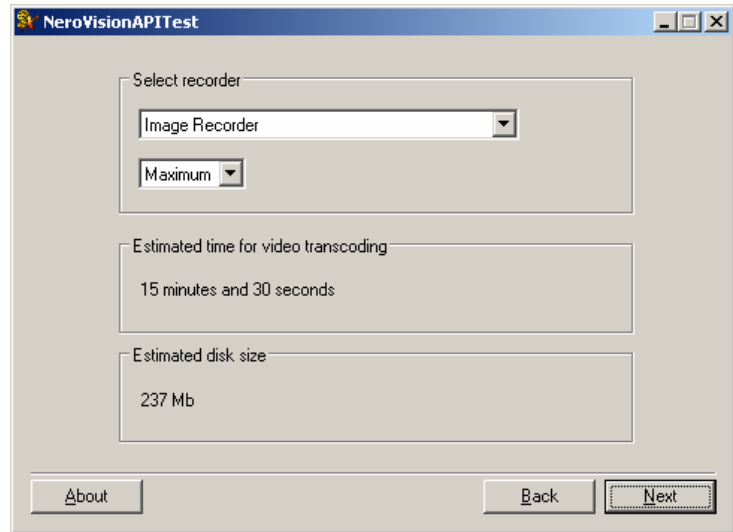
On the first page, a *NeroVision API* project can be loaded, saved and configured.

The user can choose between VCD, SVCD or DVD projects and add slide shows or videos. Also, the transition type between individual images of the slide show can be set.



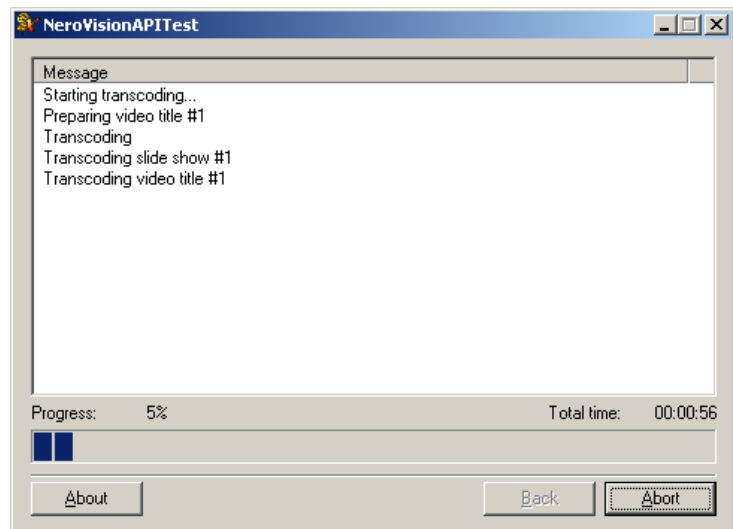
The second page offers a recorder and speed selection dialog.

The second page is the one where the *NeroVision API* is actually used to provide an estimate for transcoding time and disk size.

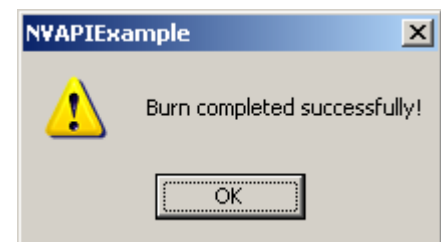


The third page provides the user with feedback on the current transcoding or burn progress.

The *NeroAPI* is used to burn the project on the desired recordable media.



Finally, the example will report the successful completion of the operation.





## 4.2. Compiling the NeroVisionAPIExample

The NeroVisionAPIExample project might not compile as intended if due to localization issues or personal preference your actual configuration differs from the default path for English versions of your Operating System.

Therefore it might be necessary to modify the source code and adjust the path the *NeroVision API* DLL before you compile the project.

Mainly, the following path in the StdAfx.h header file should be altered to match whatever location is correct in your actual environment:

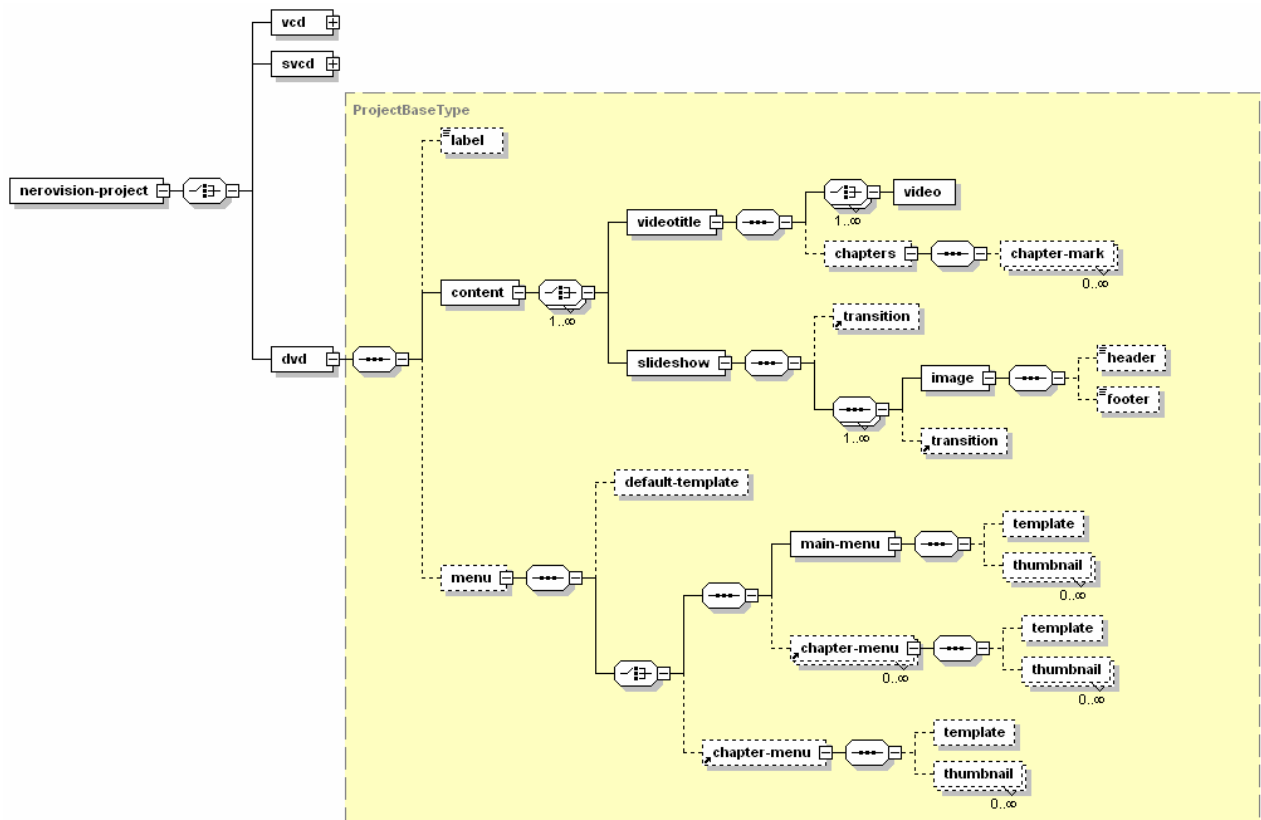
```
#import "C:\Program Files\Ahead\NeroVision\NeroVisionAPI.dll"
```

## 4.3. Accessing the *NeroVision API* in Your Applications

In general the following steps are required:

- Make the files of the *NeroAPI*-include-directory accessible from your program.
- Link your project with the NeroAPIGlue library.
- #import from the NeroVisionAPI.dll
- Use the *NeroAPI* functions in proper order.
- Implement the *NeroAPI* and *NeroVision API* callback functions.
- Create an XML project that complies with the schema definition and pass it to the *NeroVision API*
- Ask the *NeroVision API* to process the project.
- Handle the feedback from the *NeroVision API*, e.g. visualizing progress for the user.
- Use the *NeroAPI* to burn your project.

Below you can see the DVD branch of the schema completely unfolded. This shall give you an impression of the complexity you will deal with. The description of the structure as outlined by the schema is part of this documentation together with an annotated example XML project file.



#### 4.4. Points of Consideration

Make sure that the application will find the required DLLs, by installing *Nero* and *NeroVision Express*.

Provide a working infrastructure for XML processing, e.g. through MS-XML.

## 5. The NeroVisionAPIExample Application

The NeroVisionAPIExample illustrates four important aspects of working with the *NeroVision API*:

- Creating a project in XML format or loading it from hard disk
- Passing the project to the *NeroVision API* and getting estimates
- Obtaining device information through the *NeroAPI*
- Processing the project and handing it over to the *NeroAPI*

The example is a Wizard-style dialog application that leads the user through XML project creation and *NeroVision API* use with a number of tabs.

The NeroVisionAPIExample project uses MS-XML to create an XML project file that can be processed by the *NeroVision API*.

### 5.1. CAboutDlg

This is a standard about box.

### 5.2. CEstimateDoneDlg

A dialog box that holds controls showing data after transcoding disk and time estimation.

### 5.3. CEstimateDlg

A dialog box that holds controls that display progress of transcoding disk and time estimation.

### 5.4. CNVAPIExampleApp

This is a CWinApp derived class.

In InitInstance the *NeroAPI* is initialized through calls to NeroAPIGlueConnect and NeroInit. An object of the CNVAPIExampleDlg class is instantiated.

The ExitInstance function contains the code to clean up the *NeroAPI*, in particular through calls to NeroDone and NeroAPIGlueDone. Also, CoUninitialize is called.

This class also contains NeroIdleCallback and NeroUserDialog callback implementations that forward the callbacks to the currently active window and return its response.

## 5.5. CNVAPIExampleDlg

This class is an implementation of the main property sheet like dialog.

## 5.6. CNVAPIExamplePage

This template class adds a function for getting the CPageData pointer to each class that derives from it.

## 5.7. CPage

This class contains the implementation of the common page abstract class. It is not instantiated directly.

## 5.8. CPageData

This class maintains common data to be communicated among pages:

- The raw XML data string m\_sXML
- The NERO\_SCSI\_DEVICE\_INFO pointer m\_pDeviceInfo
- The burn speed value m\_dwBurnSpeed
- A pointer to the project interface in m\_pProject

The constructor creates an instance of *NeroVision API's* Project object.

## 5.9. CPageFirst

This class represents the first page in the dialog where the project is set up and contains all the XML parsing code.

The dialog allows the user to add videos and slideshows. Once a slideshow element has been created, individual images can be attached to it.

The tree items on the first level can be of the CONTENTTYPE type, where CONTENTTYPE is VIDEOTITLE or SLIDESHOW.

The tree items on the second level of a SLIDESHOW can be of the TRANSITION type. INVALID means an image path. NONE means no transition.

The two integers that hold the sequence numbers for newly inserted video and slideshow items are used for automatic unique naming of new items.

The BuildXML method is used to build an XML from the project data.

## 5.10. CPageSecond

When the second dialog is opened the API will be asked to estimate the processing time for the current project.

The central function of this class is OnChangeState.

Depending on direction from the page has been entered (coming from page one or page three), different actions will be taken.

If the page is activated for the first time, the recorder combobox is filled and the speeds combobox is updated accordingly.

If the page is activated in the forward direction, the dialog will prepare the GUI for the display of transcoding time and disk size estimations. For a potentially failed estimation it will load and set an error string.

Then the XML string as constructed on page one will be set to the *NeroVision API's* Project object. If XML setting succeeded, the application will ask for disk size and transcoding duration estimates.

Obtaining the transcoding duration estimation might take a while. Therefore, we are passing an IProgressCallback interface pointer which we are implementing. This will give us an opportunity to move the progress bar and indicate how long the process will take.

## 5.11. CPageThird

On the third page the process is started and the user is kept updated on the current progress. The currently processed item is displayed, along with the remaining time.

The central function of this class is OnStartProcess. The CreateNeroBurnContext method will not return until transcoding finishes but we have a progress callback to monitor the process. If an error occurred, a corresponding message will be displayed.

After a successful method call, finally NeroAPI's NeroBurn function will be called to perform the actual burning.

## 5.12. CRecorderCombobox

A class for a combobox that holds all available recorders.

The FillCombo function will determine the available drives through the NeroAPI and determine their characteristics and fill the combo box list accordingly.

When the control is destroyed, the device infos are freed by a call to NeroFreeMem.

The GetCurrentRecorder method will retrieve the currently selected recorder. The actual NERO\_SCSI\_DEVICE\_INFO pointer is stored in item's data pointer.

### **5.13. CSheet**

This is the common sheet abstract class.

### **5.14. CSpeedComboBox**

This is a combobox class that holds all available speeds for a particular recorder.

The RefreshSpeeds method is used to refresh the speeds of the combobox using the recorder passed as parameter. It utilizes *NeroAPI's* NeroGetAvailableSpeeds function to obtain the available speeds for the device.

## 6. An Example Project

The following is an example of an XML project file for a Super Video CD project with a slideshow containing two images, one video title and a menu.

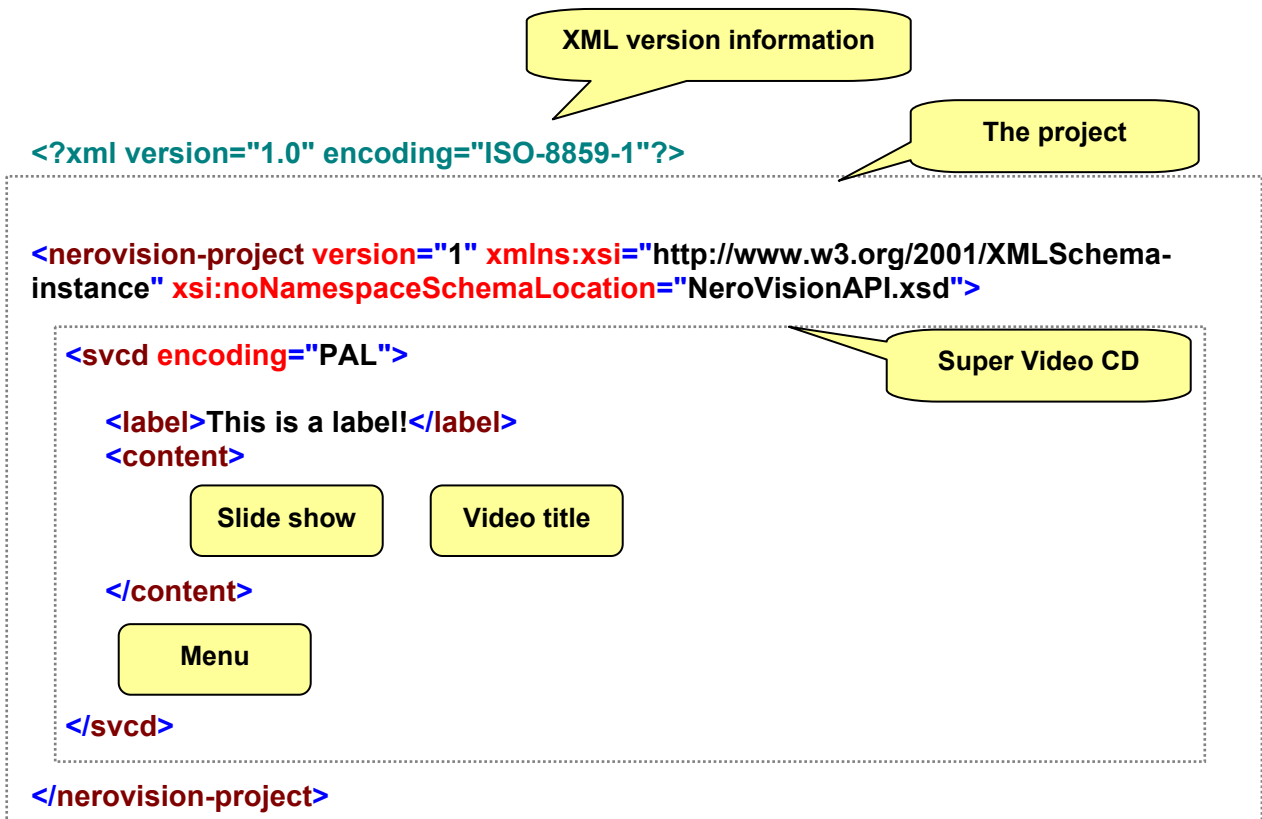
The details of the individual elements are described in the Schema Definition section of this document.



### 6.1. The Project Components

The project consists of a `nerovision-project` element which provides the frame. This project is of type "svcd" and contains a label.

The slide show and video title are embedded in the `<content>` element. The menu has its own XML element tag `<menu>`.



## 6.2. The XML File

Here is the fully expanded view of the project file. The individual segments will be explained on the following pages.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

Project Frame

```
<nerovision-project version="1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="NeroVisionAPI.xsd">
  <svcd encoding="PAL">
```

```
    <label>This is a label!</label>
    <content>
```

Slide show

```
      <slideshow name="Rome burns" id="s1">
        <transition type="WipeUp"/>
        <image src="D:\MyMediaFolder\TrulyBeautifulPicture.bmp" duration="5">
          <header>Image header</header>
        </image>
        <transition type="WipeDown"/>
        <image src="D:\MyMediaFolder\AnotherGreatPicture.bmp" duration="5">
          <header font-color="#000000" font-face="Arial" font-size="24">Header</header>
          <footer font-color="#000000">Footer</footer>
        </image>
        <transition type="WipeRight"/>
      </slideshow>
```

Video title

```
    <videotitle name="Nero fiddles" id="v1">
      <video src="D:\MyMediaFolder\GreatActionMovie.avi"/>
      <chapters>
        <chapter-mark pos="120" id="cm1" name="Interesting chapter"/>
        <chapter-mark pos="180" id="cm2" name="Interesting chapter"/>
      </chapters>
    </videotitle>
```

Menu

```
  </content>

  <menu default-thumbnail="FirstNonEmptyFrame">
    <default-template id="templ1" name="Island"/>
    <main-menu/>
    <chapter-menu title-id="v1"/>
  </menu>
```

```
</svcd>
</nerovision-project>
```



### 6.3. The Slide Show

The slide show consists of two images.

The first image will appear with a "wipe up" transition and contains a header text. This text will display in the upper part of the image in standard system font.

After 5 seconds the second image will appear with a "wipe down" transition. It contains a header with color code for black, font face Arial and a size of 24.

The footer will be black as well. The other font characteristics will be taken from the standard system font.

The second image will disappear with a "wipe right" transition after 5 seconds.

```
<slideshow name="Rome burns" id="s1">
```

```
<transition type="WipeUp"/>
```

```
<image src="D:\MyMediaFolder\TrulyBeautifulPicture.bmp" duration="5">
```

```
<header>Image header</header>
```

```
</image>
```

Transition and first image  
with header

```
<transition type="WipeDown"/>
```

```
<image src="D:\MyMediaFolder\AnotherGreatPicture.bmp" duration="5">
```

```
<header font-color="#000000" font-face="Arial" font-size="24">Header</header>
```

```
<footer font-color="#000000">Footer</footer>
```

```
</image>
```

Transition and second image  
with header and footer

```
<transition type="WipeRight"/>
```

```
</slideshow>
```

### 6.4. The Video Title

The Super Video CD project contains one video title which consists of one video file. Two chapter marks are given. The first one is 120 seconds from the start, the second one 180 seconds.

```
<videotitle name="Nero fiddles" id="v1">
```

```
<video src="D:\MyMediaFolder\GreatActionMovie.avi"/>
```

Chapter mark definition

```
<chapters>
```

```
<chapter-mark pos="120" id="cm1" name="Interesting chapter"/>
```

```
<chapter-mark pos="180" id="cm2" name="Interesting chapter"/>
```

```
</chapters>
```

```
</videotitle>
```

## 6.5. The Menu

The menu contains links to the individual chapters of the video title. By default the chapters will be represented by a thumbnail image of the first non empty frame.

```
<menu default-thumbnail="FirstNonEmptyFrame">  
  <default-template id="templ1" name="Island"/>  
  <main-menu/>  
  <chapter-menu title-id="v1"/>  
</menu>
```

Reference to template

Reference to video title

## 7. Interfaces and Types

This paragraph describes the interface to the *NeroVision API* DLL.

### 7.1. IError Interface

Error code and error description.

#### 7.1.1. ErrCode Enum

The error code is an enumeration type with the following constants:

Description of enumerators	
NoError	No error.
Canceled	Canceled by user.
InternalError	Internal error.
APIUsageError	API usage error.
InstallationError	An error with the installation.
XMLSyntaxError	Incorrect XML syntax.
FileIOError	Error during file I/O.
MenuTemplateError	Menu template error.
NeroAPIError	An error while working with the NeroAPI.
TranscodingFailed	Transcoding could not be finished successfully.
DiscOverflow	Disc overflow.

#### 7.1.2. ErrCode Property

Gets the error code.

```
HRESULT ErrCode([out, retval] ErrorCode* errCode);
```

#### 7.1.3. ErrText Property

Gets the error text.

```
HRESULT ErrText([out, retval] BSTR* errStr);
```

#### 7.1.4. XMLID Property

Gets the ID of the XML tag involved in the error.

```
HRESULT XMLID([out, retval] BSTR* xmlID);
```

### 7.2. INeroBurnContext Interface

The result of CreateNeroBurnContext.

### 7.2.1. NERO\_CD\_FORMAT

Gets the resulting NERO\_CD\_FORMAT (2. parameter of NeroBurn()).

```
HRESULT NERO_CD_FORMAT([out, retval] int* pFormat);
```

### 7.2.2. NERO\_WRITE\_CD

Gets pointer to the resulting NERO\_WRITE\_CD structure (3rd parameter of NeroBurn()).

```
HRESULT NERO_WRITE_CD([out, retval] void** pNWCD);
```

## 7.3. IProgressCallback Interface

Interface for handling progress information.

### 7.3.1. ProgressAction Enum

Action reported by ProgressCallback.

Description of enumerators	
Estimating	Estimating time for processing.
Preparing	Preparing data for transcoding.
Transcoding	Transcoding.
Generating	Generating.
Analyzing	Analyzing.

### 7.3.2. OnProgress Event

Called regularly to report progress of the current sub-task and the total function called.

```
HRESULT OnProgress([in] ProgressAction action, [in] BSTR itemname, [in] double currentRemain, [in] double currentTotal, [in] float currentFraction, [in] double totalRemain, [in] double totalTotal, [in] float totalFraction);
```

### 7.3.3. ShouldCancel Event

Called regularly to check whether the called function should be canceled.

```
HRESULT ShouldCancel([out, retval] boolean* pbCancel);
```

## 7.4. IProject Interface

A NeroVision API project describes a DVD/(S)VCD.

#### 7.4.1. SetXMLString

Sets the project content from raw XML data.

```
HRESULT SetXMLString([in] const char* data, [out, retval] boolean*
pbSuccess);
```

#### 7.4.2. SetXMLFile

Sets the project content from a XML file.

```
HRESULT SetXMLFile([in] BSTR filepath, [out, retval] boolean* pbSuccess);
```

#### 7.4.3. EstimateDiskSize

Estimates the size of the resulting disk.

```
HRESULT EstimateDiskSize([out] hyper* size, [out, retval] boolean*
pbSuccess);
```

#### 7.4.4. EstimateCreateNeroBurnContextTime

Estimates the time needed by CreateNeroBurnContext().

```
HRESULT EstimateCreateNeroBurnContextTime([in] IProgressCallback*
pCallback, [out] double* seconds, [out, retval] boolean* pbSuccess);
```

#### 7.4.5. CreateNeroBurnContext

This method creates a NeroBurnContext object which can be burned with Nero. If a ProgressCallback is passed, it will receive progress information.

```
HRESULT CreateNeroBurnContext([in] HMODULE hNeroAPIDLL, [in]
IProgressCallback* pCallback, [out] INeroBurnContext** ppburnContext,
[out, retval] boolean* pbSuccess);
```

#### 7.4.6. TempDirectory

This method sets the directory used for storing temporary files. Defaults to the system temp. directory.

```
HRESULT TempDirectory([in] BSTR path);
```

#### 7.4.7. LastError

Gets the error object of the last called function.

```
HRESULT LastError([out, retval] IError** pError);
```

### 7.5. IMediaInfo Interface

Attributes and methods for audio & video of a single media.

### 7.5.1. FileType Enum

MediaInfo file type.

Description of enumerators	
Invalid	Invalid file type.
Other	Other file type.
MPEG1	MPEG1 file.
MPEG2	MPEG2 file.
DV_DV	DV_DV file.
DV_AVI1	DV_AVI1 file.
DV_AVI2	DV_AVI2 file.

### 7.5.2. FileType Property

```
[propget, helpstring("Media file type")]
HRESULT FileType([out, retval] FileType* ft);
```

### 7.5.3. Duration Property

```
[propget, helpstring("Duration in seconds")]
HRESULT Duration([out, retval] double* sec);
```

### 7.5.4. HasAudio Property

```
[propget, helpstring("Whether the Media has audio")]
HRESULT HasAudio([out, retval] boolean* has);
```

### 7.5.5. HasVideo Property

```
[propget, helpstring("Whether the Media has video")]
HRESULT HasVideo([out, retval] boolean* has);
```

### 7.5.6. Width Property

```
[propget, helpstring("Media width")]
HRESULT Width([out, retval] int* w);
```

### 7.5.7. Height Property

```
[propget, helpstring("Media height")]
HRESULT Height([out, retval] int* h);
```

### 7.5.8. AspectXByY Property

```
[propget, helpstring("Media TV system")]
HRESULT AspectXByY([out, retval] double* aspect);
```

### 7.5.9. FrameRate Property

```
[propget, helpstring("Frame time in seconds")]
HRESULT FrameRate([out, retval] double* rate);
```

### 7.5.10. GetThumbnail Method

Retrieves the first (non-empty) thumbnail of the media.

```
HRESULT GetThumbnail([in] boolean ensureNotEmpty, [out, retval] long* hDIB);
```

### 7.5.11. GetImage Method

Retrieves a frame of the media at a given position.

```
HRESULT GetImage([in] double posSec, [out, retval] long* hDIB);
```

## 7.6. IFileInfo Interface

Attributes and methods for a physical file, which may contain several segments.

### 7.6.1. FilePath Property

```
[propget, helpstring("The associated file's path and name")]  
HRESULT FilePath([out, retval] BSTR* filepath);
```

### 7.6.2. NumSegments Property

```
[propget, helpstring("Gets the number of segments in the file")]  
HRESULT NumSegments([out, retval] int* numSeg);
```

### 7.6.3. GetSegmentMediaInfo Method

Gets the MediaInfo for one segment of the file.

```
HRESULT GetSegmentMediaInfo([in] int segIndex, [out, retval] IMediaInfo** pMediaInfo);
```

## 7.7. IMediaAnalyzer Interface

Interface for analyzing one or more media files.

### 7.7.1. AddFile Method

Add a file to be analyzed. The resulting IFileInfo is filled after calling AnalyzeFiles().

```
HRESULT AddFile([in] BSTR filepath, [out, retval] IFileInfo** fileInfo);
```

### 7.7.2. AnalyzeFiles Method

Analyzes all files previously added by AddFile().

```
HRESULT AnalyzeFiles([in] IProgressCallback* pCallback, [out, retval]  
boolean* pbSuccess);
```

### 7.7.3. LastError Property

```
[propget, helpstring("Gets the error object of the last called  
function")]  
HRESULT LastError([out, retval] IError** pError);
```

## 7.8. Project Object

An object that represents a *NeroVision API* project.

```
coclass Project  
{  
    [default] interface IProject;  
};
```

## 7.9. MediaAnalyzer Object

A Media analyzer object.

```
coclass MediaAnalyzer  
{  
    [default] interface IMediaAnalyzer;  
};
```



## 8. The XML Schema Definition

The W3C defines an XML Schema like this:

"XML Schemas express shared vocabularies and allow machines to carry out rules made by people. They provide a means for defining the structure, content and semantics of XML documents."

An XML Schema represents the rules for elements and attributes that can appear in a document, including their data types and default and fixed values. It also defines which elements are child elements, and regulates their order and number.

Schema **NeroVisionAPI.xsd**

Elements	Complex types	Simple types	Attr. groups
chapter-menu	MenuTemplateBaseType	AVTime	FontAttrGrp
nerovision-project	ProjectBaseType	FileLocation	
transition	ThumbnailBaseType	RGBColor	
		TransitionType	
		VideoEncoding	

In the following we will list the elements and attributes as they appear in the schema, also noting their uses.

### 8.1. element nerovision-project

This is the top level element of any project. A project can either be a Video CD, Super Video CD or DVD.

Though the target media have different formats the structure of the project as accepted by *NeroVision API* is identical.

You will see that the elements representing them all are of the ProjectBaseType type.

diagram						
children	vcd svcd dvd					
attributes	Name	Type	Use	Default	Fixed	Annotation
	version	xs:integer	required		1	
source	<pre> &lt;xs:element name="nerovision-project"&gt;   &lt;xs:complexType&gt;     &lt;xs:choice&gt;       &lt;xs:element name="vcd" type="ProjectBaseType"/&gt;       &lt;xs:element name="svcd" type="ProjectBaseType"/&gt;       &lt;xs:element name="dvd" type="ProjectBaseType"/&gt;     &lt;/xs:choice&gt;     &lt;xs:attribute name="version" use="required" fixed="1"&gt; </pre>					

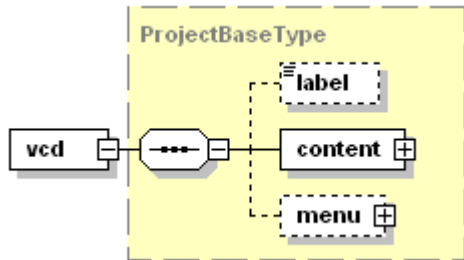
```

<xs:simpleType>
  <xs:restriction base="xs:integer">
    <xs:minExclusive value="0"/>
  </xs:restriction>
</xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>

```

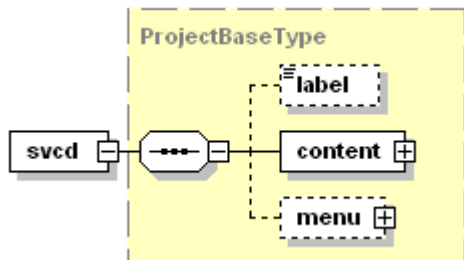
## 8.2. element nerovision-project/vcd

A Video CD project.

diagram						
type	ProjectBaseType					
children	label content menu					
attributes	Name	Type	Use	Default	Fixed	Annotation
	encoding	VideoEncoding	required			
source	<xs:element name="vcd" type="ProjectBaseType"/>					

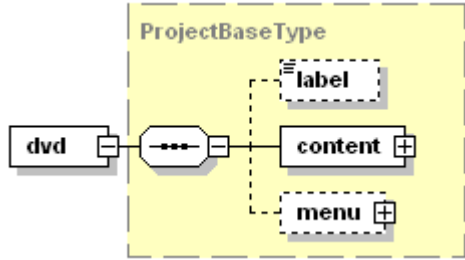
## 8.3. element nerovision-project/svcd

A Super Video CD project.

diagram						
type	ProjectBaseType					
children	label content menu					
attributes	Name	Type	Use	Default	Fixed	Annotation
	encoding	VideoEncoding	required			
source	<xs:element name="svcd" type="ProjectBaseType"/>					

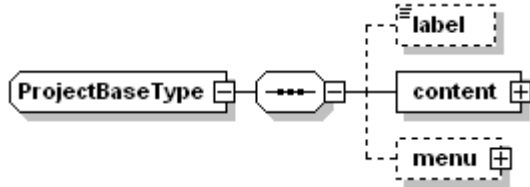
## 8.4. element nerovision-project/dvd

A DVD project.

diagram						
type	ProjectBaseType					
children	label content menu					
attributes	Name encoding	Type VideoEncoding	Use required	Default	Fixed	Annotation
source	<xs:element name="dvd" type="ProjectBaseType"/>					

## 8.5. complexType ProjectBaseType

This type represents a generic project. An actual project might be a Video CD, Super Video CD or DVD. All those are based on ProjectBaseType.

diagram						
children	label content menu					
used by	elements nerovision-project/dvd nerovision-project/svcd nerovision-project/vcd					
attributes	Name encoding	Type VideoEncoding	Use required	Default	Fixed	Annotation
source	<pre> &lt;xs:complexType name="ProjectBaseType"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="label" minOccurs="0"&gt;       &lt;xs:simpleType&gt;         &lt;xs:restriction base="xs:string"&gt;           &lt;xs:maxLength value="32"/&gt;         &lt;/xs:restriction&gt;       &lt;/xs:simpleType&gt;     &lt;/xs:element&gt;     &lt;xs:element name="content"&gt;       &lt;xs:complexType&gt;         &lt;xs:choice maxOccurs="unbounded"&gt;           &lt;xs:element name="videotitle"&gt;             &lt;xs:complexType&gt;               &lt;xs:sequence&gt;                 &lt;xs:choice maxOccurs="unbounded"&gt;                   &lt;xs:element name="video"&gt; </pre>					

```

<xs:complexType>
  <xs:attribute name="src" type="FileLocation" use="required"/>
  <xs:attribute name="id" type="xs:ID" use="optional"/>
  <xs:attribute name="start" type="AVTime" use="optional" default="0"/>
  <xs:attribute name="duration" type="AVTime" use="optional"/>
</xs:complexType>
</xs:element>
</xs:choice>
<xs:element name="chapters" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="chapter-mark" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:attribute name="pos" type="AVTime" use="required"/>
          <xs:attribute name="id" type="xs:ID" use="optional"/>
          <xs:attribute name="name" type="xs:string" use="optional"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="name" type="xs:string" use="required"/>
<xs:attribute name="id" type="xs:ID" use="optional"/>
</xs:complexType>
</xs:element>
<xs:element name="slideshow">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="transition" minOccurs="0"/>
      <xs:sequence maxOccurs="unbounded">
        <xs:element name="image">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="header" minOccurs="0">
                <xs:complexType>
                  <xs:simpleContent>
                    <xs:extension base="xs:string">
                      <xs:attributeGroup ref="FontAttrGrp"/>
                    </xs:extension>
                  </xs:simpleContent>
                </xs:complexType>
              </xs:element>
              <xs:element name="footer" minOccurs="0">
                <xs:complexType>
                  <xs:simpleContent>
                    <xs:extension base="xs:string">
                      <xs:attributeGroup ref="FontAttrGrp"/>
                    </xs:extension>
                  </xs:simpleContent>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
          <xs:attribute name="src" type="FileLocation" use="required"/>
          <xs:attribute name="duration" type="AVTime" use="required"/>
          <xs:attribute name="id" type="xs:ID" use="optional"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```


```

        <xs:element ref="transition" minOccurs="0"/>
    </xs:sequence>
</xs:sequence>
<xs:attribute name="name" type="xs:string" use="required"/>
<xs:attribute name="id" type="xs:ID" use="optional"/>
</xs:complexType>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:element>
<xs:element name="menu" minOccurs="0">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="default-template" minOccurs="0">
                <xs:complexType>
                    <xs:complexContent>
                        <xs:extension base="MenuTemplateBaseType"/>
                    </xs:complexContent>
                </xs:complexType>
            </xs:element>
            <xs:choice>
                <xs:sequence>
                    <xs:element name="main-menu">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element name="template" type="MenuTemplateBaseType" minOccurs="0"/>
                                <xs:element name="thumbnail" minOccurs="0" maxOccurs="unbounded">
                                    <xs:complexType>
                                        <xs:complexContent>
                                            <xs:extension base="ThumbnailBaseType">
                                                <xs:attribute name="title-id" type="xs:IDREF" use="required"/>
                                            </xs:extension>
                                        </xs:complexContent>
                                    </xs:complexType>
                                </xs:element>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                    <xs:element ref="chapter-menu" minOccurs="0" maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:element ref="chapter-menu" minOccurs="0"/>
            </xs:choice>
        </xs:sequence>
        <xs:attribute name="default-thumbnail" use="optional">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:enumeration value="FirstFrame"/>
                    <xs:enumeration value="FirstNonEmptyFrame"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="encoding" type="VideoEncoding" use="required"/>
</xs:complexType>

```

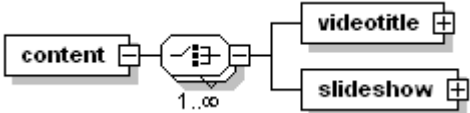
## 8.6. element ProjectBaseType/label

An optional label that can be attached to a project.

diagram	
type	restriction of <b>xs:string</b>
facets	maxLength 32
source	<pre> &lt;xs:element name="label" minOccurs="0"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:string"&gt;       &lt;xs:maxLength value="32"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:element&gt; </pre>

## 8.7. element ProjectBaseType/content

The project content can be a sequence of video titles and slideshows.

diagram	
children	videotitle slideshow
source	<pre> &lt;xs:element name="content"&gt;   &lt;xs:complexType&gt;     &lt;xs:choice maxOccurs="unbounded"&gt;       &lt;xs:element name="videotitle"&gt;         &lt;xs:complexType&gt;           &lt;xs:sequence&gt;             &lt;xs:choice maxOccurs="unbounded"&gt;               &lt;xs:element name="video"&gt;                 &lt;xs:complexType&gt;                   &lt;xs:attribute name="src" type="FileLocation" use="required"/&gt;                   &lt;xs:attribute name="id" type="xs:ID" use="optional"/&gt;                   &lt;xs:attribute name="start" type="AVTime" use="optional" default="0"/&gt;                   &lt;xs:attribute name="duration" type="AVTime" use="optional"/&gt;                 &lt;/xs:complexType&gt;               &lt;/xs:element&gt;             &lt;/xs:choice&gt;             &lt;xs:element name="chapters" minOccurs="0"&gt;               &lt;xs:complexType&gt;                 &lt;xs:sequence&gt;                   &lt;xs:element name="chapter-mark" minOccurs="0" maxOccurs="unbounded"&gt;                     &lt;xs:complexType&gt;                       &lt;xs:attribute name="pos" type="AVTime" use="required"/&gt;                       &lt;xs:attribute name="id" type="xs:ID" use="optional"/&gt;                       &lt;xs:attribute name="name" type="xs:string" use="optional"/&gt;                     &lt;/xs:complexType&gt;                   &lt;/xs:element&gt;                 &lt;/xs:sequence&gt;               &lt;/xs:complexType&gt;             &lt;/xs:element&gt;           &lt;/xs:sequence&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;     &lt;/xs:choice&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>

```

</xs:element>
</xs:sequence>
<xs:attribute name="name" type="xs:string" use="required"/>
<xs:attribute name="id" type="xs:ID" use="optional"/>
</xs:complexType>
</xs:element>
<xs:element name="slideshow">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="transition" minOccurs="0"/>
      <xs:sequence maxOccurs="unbounded">
        <xs:element name="image">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="header" minOccurs="0">
                <xs:complexType>
                  <xs:simpleContent>
                    <xs:extension base="xs:string">
                      <xs:attributeGroup ref="FontAttrGrp"/>
                    </xs:extension>
                  </xs:simpleContent>
                </xs:complexType>
              </xs:element>
              <xs:element name="footer" minOccurs="0">
                <xs:complexType>
                  <xs:simpleContent>
                    <xs:extension base="xs:string">
                      <xs:attributeGroup ref="FontAttrGrp"/>
                    </xs:extension>
                  </xs:simpleContent>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
            <xs:attribute name="src" type="FileLocation" use="required"/>
            <xs:attribute name="duration" type="AVTime" use="required"/>
            <xs:attribute name="id" type="xs:ID" use="optional"/>
          </xs:complexType>
        </xs:element>
        <xs:element ref="transition" minOccurs="0"/>
      </xs:sequence>
    </xs:sequence>
    <xs:attribute name="name" type="xs:string" use="required"/>
    <xs:attribute name="id" type="xs:ID" use="optional"/>
  </xs:complexType>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:element>

```

## 8.8. element ProjectBaseType/content/videotitle

A video title represents a collection of individual video files. This collection can be addressed by the ID for the video title. The video title can be segmented in chapters.

diagram						
children	video chapters					
attributes	Name	Type	Use	Default	Fixed	Annotation
	name	xs:string	required			
	id	xs:ID	optional			
source	<pre> &lt;xs:element name="videotitle"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:choice maxOccurs="unbounded"&gt;         &lt;xs:element name="video"&gt;           &lt;xs:complexType&gt;             &lt;xs:attribute name="src" type="FileLocation" use="required"/&gt;             &lt;xs:attribute name="id" type="xs:ID" use="optional"/&gt;             &lt;xs:attribute name="start" type="AVTime" use="optional" default="0"/&gt;             &lt;xs:attribute name="duration" type="AVTime" use="optional"/&gt;           &lt;/xs:complexType&gt;         &lt;/xs:element&gt;       &lt;/xs:choice&gt;       &lt;xs:element name="chapters" minOccurs="0"&gt;         &lt;xs:complexType&gt;           &lt;xs:sequence&gt;             &lt;xs:element name="chapter-mark" minOccurs="0" maxOccurs="unbounded"&gt;               &lt;xs:complexType&gt;                 &lt;xs:attribute name="pos" type="AVTime" use="required"/&gt;                 &lt;xs:attribute name="id" type="xs:ID" use="optional"/&gt;                 &lt;xs:attribute name="name" type="xs:string" use="optional"/&gt;               &lt;/xs:complexType&gt;             &lt;/xs:element&gt;           &lt;/xs:sequence&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>					



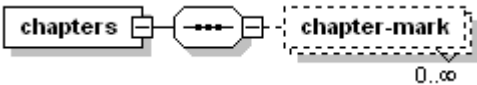
## 8.9. element ProjectBaseType/content/videotitle/video

A single video file. The ID field can be used to provide a unique identifier.

diagram	<div><div>video</div></div>																																			
attributes	<table><thead><tr><th>Name</th><th>Type</th><th>Use</th><th>Default</th><th>Fixed</th><th>Annotation</th></tr></thead><tbody><tr><td>src</td><td>FileLocation</td><td>required</td><td></td><td></td><td></td></tr><tr><td>id</td><td>xs:ID</td><td>optional</td><td></td><td></td><td></td></tr><tr><td>start</td><td>AVTime</td><td>optional</td><td>0</td><td></td><td></td></tr><tr><td>duration</td><td>AVTime</td><td>optional</td><td></td><td></td><td></td></tr></tbody></table>	Name	Type	Use	Default	Fixed	Annotation	src	FileLocation	required				id	xs:ID	optional				start	AVTime	optional	0			duration	AVTime	optional								
Name	Type	Use	Default	Fixed	Annotation																															
src	FileLocation	required																																		
id	xs:ID	optional																																		
start	AVTime	optional	0																																	
duration	AVTime	optional																																		
source	<pre>&lt;xs:element name="video"&gt;   &lt;xs:complexType&gt;     &lt;xs:attribute name="src" type="FileLocation" use="required"/&gt;     &lt;xs:attribute name="id" type="xs:ID" use="optional"/&gt;     &lt;xs:attribute name="start" type="AVTime" use="optional" default="0"/&gt;     &lt;xs:attribute name="duration" type="AVTime" use="optional"/&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>																																			


## 8.10. element ProjectBaseType/content/videotitle/chapters

A sequence of chapter marks to structure a video title.

diagram						
children	chapter-mark					
source	<pre> &lt;xs:element name="chapters" minOccurs="0"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="chapter-mark" minOccurs="0" maxOccurs="unbounded"&gt;         &lt;xs:complexType&gt;           &lt;xs:attribute name="pos" type="AVTime" use="required"/&gt;           &lt;xs:attribute name="id" type="xs:ID" use="optional"/&gt;           &lt;xs:attribute name="name" type="xs:string" use="optional"/&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>					

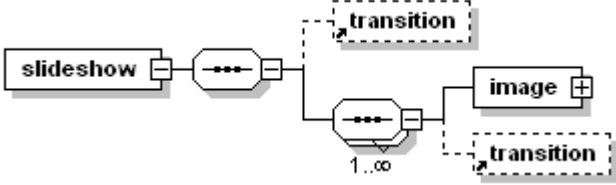
## 8.11. element ProjectBaseType/content/videotitle/chapters/chapter-mark

A chapter-mark references a significant position in a video title.

diagram						
attributes	Name	Type	Use	Default	Fixed	Annotation
	pos	AVTime	required			
	id	xs:ID	optional			
	name	xs:string	optional			
source	<pre>&lt;xs:element name="chapter-mark" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;xs:complexType&gt;     &lt;xs:attribute name="pos" type="AVTime" use="required"/&gt;     &lt;xs:attribute name="id" type="xs:ID" use="optional"/&gt;     &lt;xs:attribute name="name" type="xs:string" use="optional"/&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>					

## 8.12. element ProjectBaseType/content/slideshow

A slideshow is a sequence of images and optional transitions between them.

diagram						
children	transition image					
attributes	Name	Type	Use	Default	Fixed	Annotation
	name	xs:string	required			
	id	xs:ID	optional			
source	<pre>&lt;xs:element name="slideshow"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element ref="transition" minOccurs="0"/&gt;       &lt;xs:sequence maxOccurs="unbounded"&gt;         &lt;xs:element name="image"&gt;           &lt;xs:complexType&gt;             &lt;xs:sequence&gt;               &lt;xs:element name="header" minOccurs="0"&gt;                 &lt;xs:complexType&gt;                   &lt;xs:simpleContent&gt;                     &lt;xs:extension base="xs:string"&gt;                       &lt;xs:attributeGroup ref="FontAttrGrp"/&gt;                     &lt;/xs:extension&gt;                   &lt;/xs:simpleContent&gt;                 &lt;/xs:complexType&gt;               &lt;/xs:element&gt;               &lt;xs:element name="footer" minOccurs="0"&gt;                 &lt;xs:complexType&gt;</pre>					

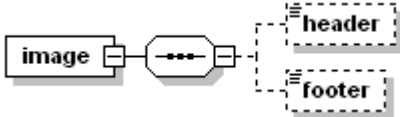
```

<xs:simpleContent>
  <xs:extension base="xs:string">
    <xs:attributeGroup ref="FontAttrGrp"/>
  </xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="src" type="FileLocation" use="required"/>
<xs:attribute name="duration" type="AVTime" use="required"/>
<xs:attribute name="id" type="xs:ID" use="optional"/>
</xs:complexType>
</xs:element>
<xs:element ref="transition" minOccurs="0"/>
</xs:sequence>
</xs:sequence>
<xs:attribute name="name" type="xs:string" use="required"/>
<xs:attribute name="id" type="xs:ID" use="optional"/>
</xs:complexType>
</xs:element>

```

### 8.13. element ProjectBaseType/content/slideshow/image

Individual element for slide shows. Every image is displayed for the time given in the "duration" attribute.

diagram						
children	header footer					
attributes	Name	Type	Use	Default	Fixed	Annotation
	src	FileLocation	required			
	duration	AVTime	required			
	id	xs:ID	optional			
source	<pre> &lt;xs:element name="image"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="header" minOccurs="0"&gt;         &lt;xs:complexType&gt;           &lt;xs:simpleContent&gt;             &lt;xs:extension base="xs:string"&gt;               &lt;xs:attributeGroup ref="FontAttrGrp"/&gt;             &lt;/xs:extension&gt;           &lt;/xs:simpleContent&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;       &lt;xs:element name="footer" minOccurs="0"&gt;         &lt;xs:complexType&gt;           &lt;xs:simpleContent&gt;             &lt;xs:extension base="xs:string"&gt;               &lt;xs:attributeGroup ref="FontAttrGrp"/&gt;             &lt;/xs:extension&gt;           &lt;/xs:simpleContent&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>					


```

</xs:element>
</xs:sequence>
<xs:attribute name="src" type="FileLocation" use="required"/>
<xs:attribute name="duration" type="AVTime" use="required"/>
<xs:attribute name="id" type="xs:ID" use="optional"/>
</xs:complexType>
</xs:element>

```


## 8.14. element ProjectBaseType/content/slideshow/image/header

An optional header text for an image.

diagram						
type	extension of <b>xs:string</b>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	font-face	xs:string	optional			
	font-color	RGBColor	optional			
	font-size	xs:int	optional			
source	<pre> &lt;xs:element name="header" minOccurs="0"&gt;   &lt;xs:complexType&gt;     &lt;xs:simpleContent&gt;       &lt;xs:extension base="xs:string"&gt;         &lt;xs:attributeGroup ref="FontAttrGrp"/&gt;       &lt;/xs:extension&gt;     &lt;/xs:simpleContent&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>					

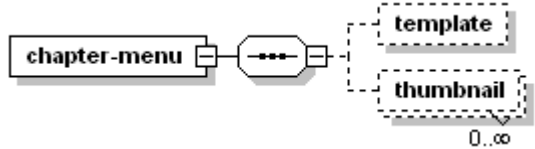
## 8.15. element ProjectBaseType/content/slideshow/image/footer

An optional footer text for an image.

diagram						
type	extension of <b>xs:string</b>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	font-face	xs:string	optional			
	font-color	RGBColor	optional			
	font-size	xs:int	optional			
source	<pre> &lt;xs:element name="footer" minOccurs="0"&gt;   &lt;xs:complexType&gt;     &lt;xs:simpleContent&gt;       &lt;xs:extension base="xs:string"&gt;         &lt;xs:attributeGroup ref="FontAttrGrp"/&gt;       &lt;/xs:extension&gt;     &lt;/xs:simpleContent&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>					

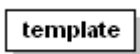
## 8.16. element chapter-menu

The chapter-menu element represents a single page that links to individual chapters of a video title. The chapter-menu can have a template of its own and a number of thumbnails.

diagram						
children	template thumbnail					
used by	elements <b>ProjectBaseType/menu</b>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	title-id	xs:IDREF	required			
source	<pre> &lt;xs:element name="chapter-menu"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="template" minOccurs="0"&gt;         &lt;xs:complexType&gt;           &lt;xs:complexContent&gt;             &lt;xs:extension base="MenuTemplateBaseType"/&gt;           &lt;/xs:complexContent&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;       &lt;xs:element name="thumbnail" minOccurs="0" maxOccurs="unbounded"&gt;         &lt;xs:complexType&gt;           &lt;xs:complexContent&gt;             &lt;xs:extension base="ThumbnailBaseType"&gt;               &lt;xs:attribute name="chapter-id" type="xs:IDREF" use="required"/&gt;             &lt;/xs:extension&gt;           &lt;/xs:complexContent&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;     &lt;xs:attribute name="title-id" type="xs:IDREF" use="required"/&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>					

## 8.17. element chapter-menu/template

The template element of chapter-menu represents the background and button style.

diagram						
type	extension of <b>MenuTemplateBaseType</b>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	id	xs:ID	optional			
	src	xs:string	optional			
	group	xs:string	optional			
	name	xs:string	required			
source	<pre> &lt;xs:element name="template" minOccurs="0"&gt;   &lt;xs:complexType&gt; </pre>					

```

<xs:complexContent>
  <xs:extension base="MenuTemplateBaseType"/>
</xs:complexContent>
</xs:complexType>
</xs:element>

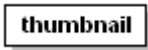
```

## 8.18. element chapter-menu/thumbnail

The thumbnail element of chapter-menu represents a small image to facilitate orientation for the user.


The source of the thumbnail can either be an image file or a single picture frame taken from a video title at a certain position which is identified by providing the AVTime.

The thumbnail must link to an existing chapter-id.

diagram						
type	extension of <b>ThumbnailBaseType</b>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	thumbnail-image-src	FileLocation	optional			
	thumbnail-posx	AVTime	optional			
	chapter-id	xs:IDREF	required			
source	<pre> &lt;xs:element name="thumbnail" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;xs:complexType&gt;     &lt;xs:complexContent&gt;       &lt;xs:extension base="ThumbnailBaseType"&gt;         &lt;xs:attribute name="chapter-id" type="xs:IDREF" use="required"/&gt;       &lt;/xs:extension&gt;     &lt;/xs:complexContent&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>					

## 8.19. element transition

The *NeroVision API* optionally allows separating the individual pictures in a slideshow by transitions. The transitions determine how images appear and disappear, e.g. with a wiping effect.

diagram						
used by	elements <b>ProjectBaseType/content/slideshow ProjectBaseType/content/slideshow</b>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	type	TransitionType	required			
	id	xs:ID	optional			
source	<pre> &lt;xs:element name="transition"&gt;   &lt;xs:complexType&gt;     &lt;xs:attribute name="type" type="TransitionType" use="required"/&gt;     &lt;xs:attribute name="id" type="xs:ID" use="optional"/&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>					

## 8.20. complexType MenuTemplateBaseType


This type represents a *NeroVision Express 2* menu template.

If "src" has been provided, this will be interpreted as a file location. Without "src" the standard *NeroVision Express 2* templates (if installed) will be used.

"group" and "name" are the optional group and required name of the template to be used from the "src" template file.

If no group information has been provided, the first template with matching name will be used.

If the requested template does not exist, the *NeroVision API* will return an error. The error object will contain the XML-ID of the offending XML element - if the optional ID was provided beforehand.

diagram						
used by	elements <b>ProjectBaseType/menu/default-template</b> <b>chapter-menu/template</b> <b>ProjectBaseType/menu/main-menu/template</b>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	id	xs:ID	optional			
	src	xs:string	optional			
	group	xs:string	optional			
	name	xs:string	required			
source	<pre> &lt;xs:complexType name="MenuTemplateBaseType"&gt;   &lt;xs:attribute name="id" type="xs:ID" use="optional"/&gt;   &lt;xs:attribute name="src" type="xs:string" use="optional"/&gt;   &lt;xs:attribute name="group" type="xs:string" use="optional"/&gt;   &lt;xs:attribute name="name" use="required"/&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:string"&gt;       &lt;xs:minLength value="1"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:complexType&gt; </pre>					

## 8.21. element ProjectBaseType/menu

A menu consists either of

- a hierarchical main-menu with a number of chapter-menus as sub menus or a chapter-menu or
- a single chapter-menu

Slide shows do not have chapter-menus.


The default-thumbnail attribute should be set to FirstFrame or FirstNonEmptyFrame.

diagram						
children	default-template main-menu chapter-menu					
attributes	Name	Type	Use	Default	Fixed	Annotation
	default-thumbnail	xs:string	optional			
source	<pre> &lt;xs:element name="menu" minOccurs="0"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="default-template" minOccurs="0"&gt;         &lt;xs:complexType&gt;           &lt;xs:complexContent&gt;             &lt;xs:extension base="MenuTemplateBaseType"/&gt;           &lt;/xs:complexContent&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;       &lt;xs:choice&gt;         &lt;xs:sequence&gt;           &lt;xs:element name="main-menu"&gt;             &lt;xs:complexType&gt;               &lt;xs:sequence&gt;                 &lt;xs:element name="template" type="MenuTemplateBaseType" minOccurs="0"/&gt;                 &lt;xs:element name="thumbnail" minOccurs="0" maxOccurs="unbounded"&gt;                   &lt;xs:complexType&gt;                     &lt;xs:complexContent&gt;                       &lt;xs:extension base="ThumbnailBaseType"&gt;                         &lt;xs:attribute name="title-id" type="xs:IDREF" use="required"/&gt;                       &lt;/xs:extension&gt;                     &lt;/xs:complexContent&gt;                   &lt;/xs:complexType&gt;                 &lt;/xs:element&gt;               &lt;/xs:sequence&gt;             &lt;/xs:complexType&gt;           &lt;/xs:element&gt;           &lt;xs:element ref="chapter-menu" minOccurs="0" maxOccurs="unbounded"/&gt;         &lt;/xs:sequence&gt;         &lt;xs:element ref="chapter-menu" minOccurs="0"/&gt;       &lt;/xs:choice&gt;     &lt;/xs:sequence&gt;     &lt;xs:attribute name="default-thumbnail" use="optional"&gt;       &lt;xs:simpleType&gt;         &lt;xs:restriction base="xs:string"&gt;           &lt;xs:enumeration value="FirstFrame"/&gt;           &lt;xs:enumeration value="FirstNonEmptyFrame"/&gt;         &lt;/xs:restriction&gt;       &lt;/xs:simpleType&gt;     &lt;/xs:attribute&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>					



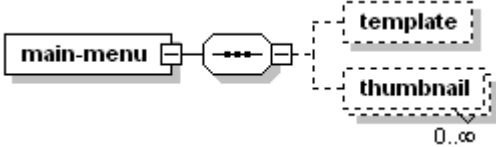
## 8.22. element ProjectBaseType/menu/default-template

The optional default-template will be used when no template element is provided for the individual menu elements.

diagram						
type	extension of <b>MenuTemplateBaseType</b>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	id	xs:ID	optional			
	src	xs:string	optional			
	group	xs:string	optional			
	name	xs:string	required			
source	<pre>&lt;xs:element name="default-template" minOccurs="0"&gt;   &lt;xs:complexType&gt;     &lt;xs:complexContent&gt;       &lt;xs:extension base="MenuTemplateBaseType"/&gt;     &lt;/xs:complexContent&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>					


## 8.23. element ProjectBaseType/menu/main-menu

Typically, a main-menu is used when there is more than one video title or a combination of video titles and slide shows.

diagram	
children	template thumbnail
source	<pre>&lt;xs:element name="main-menu"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="template" type="MenuTemplateBaseType" minOccurs="0"/&gt;       &lt;xs:element name="thumbnail" minOccurs="0" maxOccurs="unbounded"&gt;         &lt;xs:complexType&gt;           &lt;xs:complexContent&gt;             &lt;xs:extension base="ThumbnailBaseType"&gt;               &lt;xs:attribute name="title-id" type="xs:IDREF" use="required"/&gt;             &lt;/xs:extension&gt;           &lt;/xs:complexContent&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>


## 8.24. element ProjectBaseType/menu/main-menu/template

The optional template for the main menu.

diagram						
type	MenuTemplateBaseType					
attributes	Name	Type	Use	Default	Fixed	Annotation
	id	xs:ID	optional			
	src	xs:string	optional			
	group	xs:string	optional			
	name	xs:string	required			
source	<xs:element name="template" type="MenuTemplateBaseType" minOccurs="0"/>					

## 8.25. element ProjectBaseType/menu/main-menu/thumbnail


A representation for a video chapter or slide show in the main menu.

diagram						
type	extension of ThumbnailBaseType					
attributes	Name	Type	Use	Default	Fixed	Annotation
	thumbnail-image-src	FileLocation	optional			
	thumbnail-posx	AVTime	optional			
	title-id	xs:IDREF	required			
source	<pre>&lt;xs:element name="thumbnail" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;xs:complexType&gt;     &lt;xs:complexContent&gt;       &lt;xs:extension base="ThumbnailBaseType"&gt;         &lt;xs:attribute name="title-id" type="xs:IDREF" use="required"/&gt;       &lt;/xs:extension&gt;     &lt;/xs:complexContent&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>					

## 8.26. complexType ThumbnailBaseType

A thumbnail represents a chapter and can be created from an image file or a position in a video file.

The time position is relative to the start of the video title.

diagram						
used by	elements chapter-menu/thumbnail ProjectBaseType/menu/main-menu/thumbnail					
attributes	Name	Type	Use	Default	Fixed	Annotation
	thumbnail-image-src	FileLocation	optional			
	thumbnail-posx	AVTime	optional			

source	<pre>&lt;xs:complexType name="ThumbnailBaseType"&gt;   &lt;xs:attribute name="thumbnail-image-src" type="FileLocation" use="optional"/&gt;   &lt;xs:attribute name="thumbnail-posx" type="AVTime" use="optional"/&gt; &lt;/xs:complexType&gt;</pre>
--------	---

## 8.27. simpleType AVTime

Time measured in seconds. This type is used for durations as well as relative positions in video titles.

type	restriction of <b>xs:decimal</b>
used by	attributes <b>ProjectBaseType/content/videotitle/video/@duration</b> <b>ProjectBaseType/content/slideshow/image/@duration</b> <b>ProjectBaseType/content/videotitle/chapters/chapter-mark/@pos</b> <b>ProjectBaseType/content/videotitle/video/@start ThumbnailBaseType/@thumbnail-posx</b>
facets	minInclusive 0
source	<pre>&lt;xs:simpleType name="AVTime"&gt;   &lt;xs:restriction base="xs:decimal"&gt;     &lt;xs:minInclusive value="0"/&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;</pre>

## 8.28. simpleType FileLocation

The location of a video or image.

type	<b>xs:anyURI</b>
used by	attributes <b>ProjectBaseType/content/videotitle/video/@src</b> <b>ProjectBaseType/content/slideshow/image/@src</b> <b>ThumbnailBaseType/@thumbnail-image-src</b>
source	<pre>&lt;xs:simpleType name="FileLocation"&gt;   &lt;xs:restriction base="xs:anyURI"/&gt; &lt;/xs:simpleType&gt;</pre>

## 8.29. simpleType RGBColor

Used to determine font color for headers and footers.

The intensity of the red, green and blue channels (RGB) is given in hexadecimal numbers, formatted as #<red><green><blue>.

For example, #FFFFFF is white, #FF0000 red or #000000 black.

type	restriction of <b>xs:string</b>
used by	attribute <b>FontAttrGrp/@font-color</b>
facets	pattern <b>#[0-9A-Fa-f]{6}</b>
source	<pre>&lt;xs:simpleType name="RGBColor"&gt;   &lt;xs:restriction base="xs:string"&gt;     &lt;xs:pattern value="#[0-9A-Fa-f]{6}" /&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;</pre>

## 8.30. simpleType TransitionType

Available transitions between individual images in slide shows.

<b>type</b>	restriction of <b>xs:string</b>
<b>used by</b>	attribute <b>transition/@type</b>
<b>facets</b>	enumeration WipeUp enumeration WipeDown enumeration WipeLeft enumeration WipeRight enumeration MoveIn enumeration MoveOut
<b>source</b>	<pre> &lt;xs:simpleType name="TransitionType"&gt;   &lt;xs:restriction base="xs:string"&gt;     &lt;xs:enumeration value="WipeUp"/&gt;     &lt;xs:enumeration value="WipeDown"/&gt;     &lt;xs:enumeration value="WipeLeft"/&gt;     &lt;xs:enumeration value="WipeRight"/&gt;     &lt;xs:enumeration value="MoveIn"/&gt;     &lt;xs:enumeration value="MoveOut"/&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; </pre>

## 8.31. simpleType VideoEncoding

This is a region-dependent setting for the main television systems with the most significant difference being the frame rate.

PAL (Phase Alternating Line) produces 25 frames per second and NTSC (National Television System Committee) approximately 30 frames per second.

The resolution also differs slightly.

<b>type</b>	restriction of <b>xs:string</b>
<b>used by</b>	attribute <b>ProjectBaseType/@encoding</b>
<b>facets</b>	enumeration PAL enumeration NTSC
<b>source</b>	<pre> &lt;xs:simpleType name="VideoEncoding"&gt;   &lt;xs:restriction base="xs:string"&gt;     &lt;xs:enumeration value="PAL"/&gt;     &lt;xs:enumeration value="NTSC"/&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt; </pre>

## 8.32. attributeGroup FontAttrGrp

This attribute group bundles font characteristics for headers and footers of individual images in slide shows.

used by	elements <b>ProjectBaseType/content/slideshow/image/footer</b> <b>ProjectBaseType/content/slideshow/image/header</b>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	font-face	xs:string	optional			
	font-color	RGBColor	optional			
	font-size	xs:int	optional			
source	<pre> &lt;xs:attributeGroup name="FontAttrGrp"&gt;   &lt;xs:attribute name="font-face" use="optional"&gt;     &lt;xs:simpleType&gt;       &lt;xs:restriction base="xs:string"&gt;         &lt;xs:minLength value="1"/&gt;         &lt;xs:whiteSpace value="replace"/&gt;       &lt;/xs:restriction&gt;     &lt;/xs:simpleType&gt;   &lt;/xs:attribute&gt;   &lt;xs:attribute name="font-color" type="RGBColor" use="optional"/&gt;   &lt;xs:attribute name="font-size" use="optional"&gt;     &lt;xs:simpleType&gt;       &lt;xs:restriction base="xs:int"&gt;         &lt;xs:minExclusive value="0"/&gt;       &lt;/xs:restriction&gt;     &lt;/xs:simpleType&gt;   &lt;/xs:attribute&gt; &lt;/xs:attributeGroup&gt; </pre>					

## 9. Known Limitations

The limitations of the *NeroVision API* correspond to those of the *NeroAPI*:

- Currently there are no Linux versions of the *NeroVision API* and the *Nero Software Development Kit (NeroSDK)*.
- Methods of the *NeroAPI* and *NeroVision API* should not be called simultaneously from different threads.
- Only one recorder can be accessed at a time.
- Individual thumbnails for menu items are already defined in the XML Schema but currently not fully supported by the *NeroVision API*.  
Please use the “default-thumbnail” attribute of the <menu> element.