Eyeblaster Platform
Video Guide

Document Version 1.2
July 2006

For additional information, please contact our client services at
support@eyeblaster.com

Version Management

Version	Changes
1.1	Addition of screen shots.
1.2	New full screen component, API, Full screen implementation guidelines.

Table of Contents

Version Management ii
Video Overview 4
How to create a video ad 4
Creating the Video Asset 4
Package View 5
Creating a video package in the system 5
What is a Video Package? 5
Edit Package 6
Video Studio 7
Selecting Video Formats 7
How to select the right video creation track 8
  Track 1 - Create Video Package Automatically 8
  Track 2 - Build Video Package Manually 11
  Track 3 - Upload raw video asset(s) 12
  Track 4 - Send via Regular Mail to Eyeblaster 13
Working in Flash 15
  MXP Installation 15
  Video Components 17
  How to Create a Full Screen Skin 17
Full Screen Video Implementation 18
New Video Features 18
Eyeblaster Video Studio 2006 Components 20
Editing the Video Components 23
Editing Eyeblaster Control Buttons 28
Defining the Loading Animation 33
How to test the video file 33
Eyeblaster Video Studio 2006 API 34
Setting up the Ad on the Eyeblaster Platform 42
Video Enhancement Guidelines 44
Advanced Topics 45
Content Delivery Methods 45
FAQ 46

Video Overview

The Eyeblaster Platform offers a user-friendly way to handle video assets, upload and encode them and finally attach them to ads. The new Video Studio allows you to upload and encode your video assets using one of four encoding tracks best suited for your needs. You can thus easily create video content from a full range of creative formats. This ensures that you create assets most suitable for all user bandwidths and browsers. Flash runtime skins that support both regular viewing and full screen viewing can be created and easily customized within the Eyeblaster video Flash components. Bandwidth is determined by the automatic bandwidth detection tool resulting is an optimal user experience - the user is served the best quality video for their bandwidth. The full-range of user interaction metrics are also captured for reporting purposes.

How to create a video ad

Eyeblaster provides the ability to easily create video ads that ensure an optimal user experience. This process involves the creation of the video asset, creating a video package, ad creation using the Eyeblaster components, and finally setting up the assets on the Eyeblaster Platform.

The steps below provide a detailed description of the above procedure:

- Creating the Video Asset
- Creating a video package in the system
- Creating video control skins in Flash
- Setting up the ad on the Eyeblaster Platform

Creating the Video Asset

The first step for creating a successful video ad is creating the video asset. It is possible to create a unique video asset for the web, or repurpose TV ads by encoding and optimizing the video for the web.

Eyeblaster supports video playback of FLV 8, FLV 7 and WMV. Video files can be up to 4 MB (video plus ad up to 4.4MB) and support full screen playback. It is possible to encode the files directly into one of these formats, or to use Eyeblaster Video Services who allows you to encode a file from almost any raw format into the necessary formats. If you are encoding the file yourself, it is recommended to create a WMV file in the best quality possible that will fit in 2.2 MB or 4 MB. Later, when the video asset is uploaded to the Eyeblaster platform, an automatic process is performed which re-encodes the video asset to multiple formats and multiple bitrates. Since WMV can be converted to FLV (but not vice versa), it is recommended to create your initial video asset in WMV. Note that it is possible to use SWF for video, but it will not work within the Video Studio.

When creating a video asset there are a number of issues that require special attention. You need to familiarize yourself with basic concepts and preferred video creation methods. The following topics link to essential information on the Web:

- Flash Video Features.
- Windows Media Features.
- Delivery Options: It is possible to design a different video for streaming than for progressive download but with the Eyeblaster Video Component, the same asset can be played back in both modes. To learn more about streaming and progressive download, see the Streaming vs progressive download section. Choose progressive download in the component; this way the system will serve the ad in either way according to publisher/network limitations.
- Video Parameters and Standards - Most publishers accept videos of up to 30 seconds, but some allow longer videos. Standard formats for web playback are FLV (Flash Video) and WMV (Windows Media Video).
- Video Encoding

Once the video asset is prepared, it is time to create a video package, which is a set of assets all representing the same video creative, in different formats (WMV, FLV 8 & FLV 7) in multiple bitrates, which will create an optimized user experience in the best possible quality.

Package View

The Package View dialog is the dialog you see when you enter the Video Studio. It contains a list of the video packages that have been created. From this dialog you can filter, preview, edit, delete and copy your uploaded packages. This includes the ability to replace assets that you have already encoded in a package. In addition, you can access the package creation wizard directly in order to add new packages by clicking the **Add New** button.

Creating a video package in the system

An important way to optimize user experience is to create different video assets for different bitrates. Eyeblaster simplifies this process with the concept of a video package. The video package is a group of video assets all representing the same video creative. Each asset can be in a different format (WMV, FLV 8 or FLV 7) and a different bitrate. WMV and FLV 8 are higher quality video formats, but are less common than FLV 7; therefore it is important to have the ability to automatically create all formats, and provide the user with the best possible user experience depending on the plug-ins that they have installed, and their available bandwidth. Eyeblaster offers the ability to simplify the process by automatically creating all the different bitrates and formats for a single source file. All these assets are packaged together and used in a single ad which greatly simplifies the ad setup process and reduces the need to create separate ads.

The Video Studio includes a wizard that allows you to upload a video asset and process it for optimal user experience. The first step is to choose the most suitable track for uploading and encoding the asset. The tracks process your assets for suitable "package" creation. This package contains all the relevant file formats encoded according to your bitrate and compression preferences.

Check the radio button next to the video creation track you have chosen:

- Create Video Package Automatically
- Create Video Package Manually
- Upload by FTP for Encoding by Eyeblaster
- Send via Regular Mail to Eyeblaster

What is a Video Package?

A package is a collection of different versions of the same video in different formats (WMV, FLV 8 and FLV 7) and different bitrates designed to cover the various Bandwidth requirements of the user. The package replaces a single video thus allowing the system to serve the user the most suitable resource according to the user's bandwidth and the plugins they have installed.

A package may contain one or more different formats and/or one or more different bitrates. The creation of this package is enabled by first choosing the appropriate package creation track in the Video Studio and performing the various steps within the selected track.

Edit Package

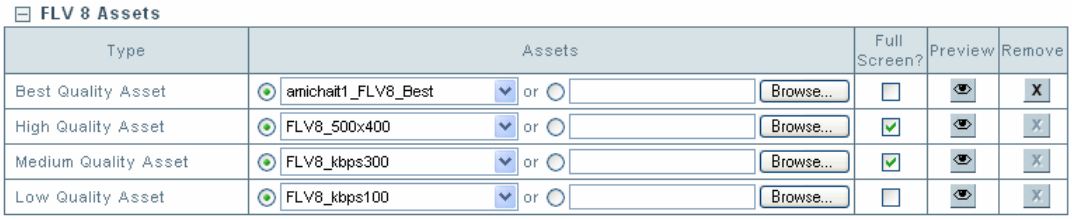
The Edit Package dialog provides extensive editing capabilities to packages uploaded to the system. These editing functions include:

- **Replacing an Asset**

  You can replace any asset either by selecting it from a folder in the drop-down list or uploading assets from your local disk. The assets must be the same format and length. The bitrate must The bitrate must not exceed the bitrate of the higher level or be below the lower level. For example, if you are replacing a medium bandwidth asset, and high is at 500 Kbps and low is at 100 Kbps, your new asset must be between 100 and 500 Kbps.

  If you replace the best bandwidth version you will be alerted whether you want to re-encode the package. If you replace the best bandwidth FLV and a WMV already exists in the package, only the FLV assets are replaced in the reencoding process. If you replace the best bandwidth WMV, all assets will be replaced in the reencoding process.

  The screen capture below shows the **FLV 8 Assets** section of the **Edit Package** dialog:



- **Deleting an Asset**

  To delete an asset, click the delete button  in the last column of the row. You cannot delete the best bandwidth version of an asset.

- **Enabling Full Screen**

  Check the **Full Screen** checkbox to enable full screen functionality for a particular asset. Note that the component must be set up properly in order for full screen to work.

- **Previewing the Asset**

  Click the **Preview** button to open the asset in the preview window. The preview includes the defined video controls which you can interact with.

- **Changing Bitrates**

  You can change the bitrate settings by clicking **Encoding settings** and entering new values. Any change to the setting will cause a re-encoding of the package. Note that this option is not available for video packages created manually.

- **Changing Formats**

  Select either FLV or WMV as the video format. Note that WMV does not support running a top layer Flash.

Click **Save** to save your changes and return to the Package List in the **Video Studio**. Any new encoding that needs to be done will now be performed. To exit without saving any of your changes, click **Cancel.**

Note: You cannot replace the assets or change the full screen setting of a package that is part of an approved or published ad.

Video Studio

The video studio provides a wizard for video package creation. This wizard offers four different tracks for creating video from which you choose the one best suited for the kind of resources you have and your particular serving requirements.

See the section How to create a video ad using the Video Studio package creation wizard.

Selecting Video Formats

Track 1 - Create Video Package Automatically

For automatic video package creation, choose either a WMV or FLV. If you choose to upload a WMV, it is possible to create different WMV versions and/or FLV files. If you choose FLV, it automatically supports both version 7 and 8 but not WMV. Upload the file with the maximum bitrate for serving the ad and no larger than 4 MB.

Note that if you are using WMV and have an “interactive on top layer”, those users that receive the WMV file will not receive the on top layer. Please check with the agency that this is an acceptable experience. If it is not, set the encoding settings so that only FLV 7 and FLV 8 are created.

If you wish the ad to display as an FLV in regular size with interactive on top layer, and as a WMV in full screen, contact Eyeblaster so they can define an advanced setting to enable this capability.

Track 2- Create Video Package Manually

For manual package creation, you actually decide what will be the contents of the package. You have to provide all the required versions of the file format you upload. As an example, it is recommended to upload both version 7 and 8 of your FLV.

You also have to provide each file format in each different bitrate needed. The maximum number of bitrate versions is four but you do not have to provide all four.

The maximum number of files that can be uploaded is 12. Three different versions and four different bitrates.

Track 3 - Upload by FTP for Encoding by Eyeblaster

For uploading for encoding by Eyeblaster, all supported formats can be uploaded such as AVI, MOV, MPEG, FLV, WMV etc. with a maximum file size of 120Mb.

Track 4 - Send via Regular Mail to Eyeblaster

For sending by regular mail, you can include all supported formats recorded on various data storage media such as CD or beta tapes.

Note: In all the tracks except for the manual track, you can determine the bitrate you want the system to encode your assets in. Eyeblaster uses default bitrates which can be customized.

How to select the right video creation track

The guiding principal for selecting the right track is to choose one that will produce a package according to the size and type of video source files you have.

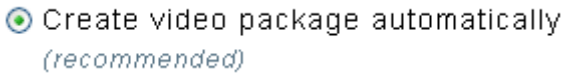
If you have a high quality file (FLV, WMV) with a maximum size of 4 MB that you want to be encoded by Eyeblaster then the Create Video Package Automatically track is most suitable for you. The package is produced in a few minutes using your high quality file to make all the appropriate assets for the appropriate bandwidths and formats. You can upload up to 5 assets to create 5 packages.

If you have several high quality files (FLV, WMV) with a maximum size of 4 MB that you have already encoded different bandwidths and formats then the Create Video Package Manually track is most suitable. It lets you manually provide all the file formats and bitrates that make up the package. The package is assembled instantly.

If your files are not encoded yet and are still in raw digital format, MOV, AVI, etc., and have a maximum size of 120 Mb, then the Upload raw video asset(s) for encoding by Eyeblaster track provides such an encoding solution. The estimated delivery time is between 2-3 business days.

If your files are greater than 120 MB or are "non-digitized" for example, DVD or a beta tape, then the Send raw video asset(s) by postal mail for encoding track provides a solution for submitting your files by mail and have them encoded by Eyeblaster. The estimated delivery time is between 5-7 business days once the media has been received. You can create up to 5 packages for sending.

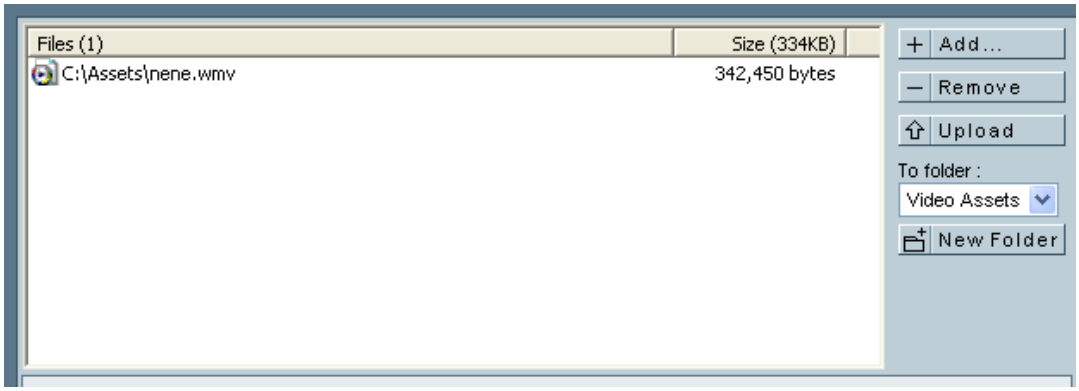
Track 1 - Create Video Package Automatically



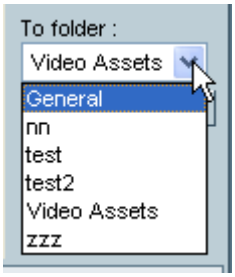
Upload (Skip this step if you have already uploaded the file on the creative assets page)

1. Click  to open the dialog for handling the assets you want to upload to the system.

2. Click  **Add** and browse to the FLV or WMV you want to upload. You may select more than one file at a time. The selected file(s) will appear in the list:



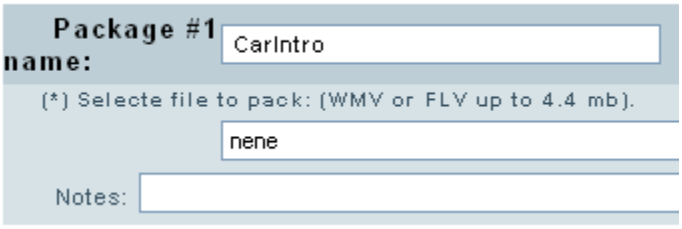
3. From the To folder drop-down list select the folder in which you want to place the file:



4. Click **New Folder** and enter the name of a new folder if you need one.
5. Click **Upload** to upload them to the system. Remember that you must select either a WMV or FLV with a maximum size of 4 MB. (Do not close the progress bar window until the asset has finished uploading or this will stop the uploading process).

Select Content

6. Enter a package name. Choose a name that will enable you to identify the package from other packages in the additional assets dialog when creating the ad.

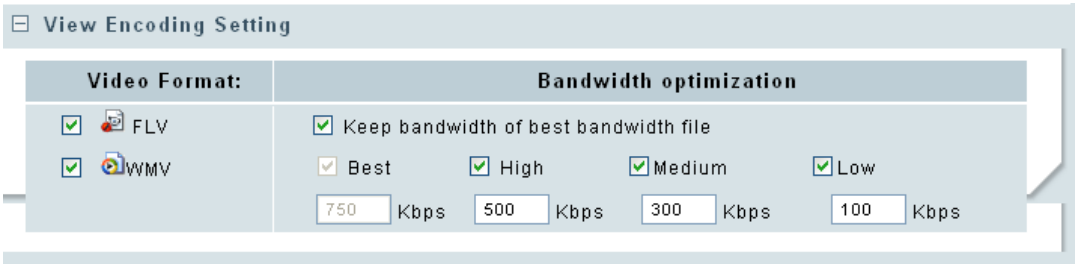


7. Select file that was already uploaded to the system from the drop-down menu:

8. Enter free text in the **Notes** field. This is typically information for the use of creative shops.

View Encoding Settings (Optional)

Expand the **View Encoding Settings** section if you want to change the default settings. These default settings are the recommended ones:



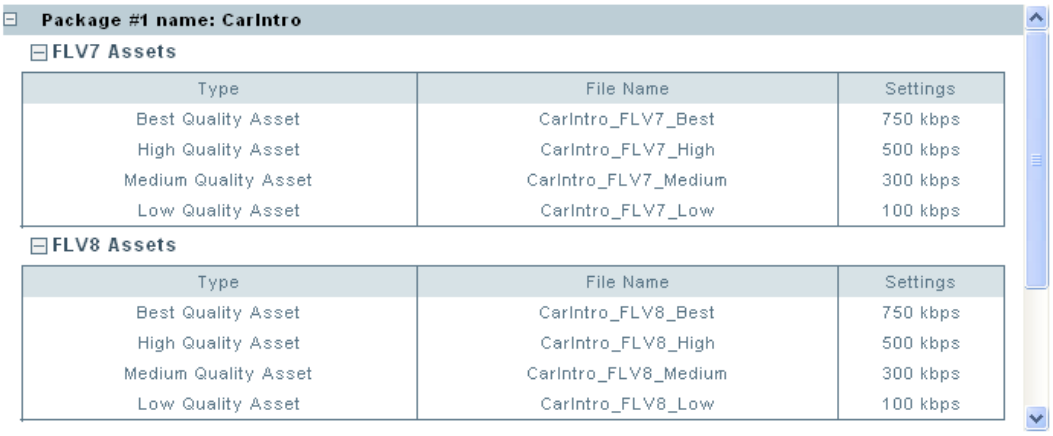
9. Check the checkbox next to the video formats you wish to create. (FLV, WMV or both). Note that if your source file is WMV you can create both WMV and FLV whereas if you have only FLV as your source file, than you can create only FLV.
10. Check the **Bandwidth** checkboxes and, if you want, change the default bitrate for **Best**, **High**, **Medium** and **Low** bandwidths. Note that the default bitrates are based on the statistics from actual user bandwidths and are therefore the recommended values. If **Best** is less than 750 Kbps, the rest of the bitrates will be adjusted and divided evenly at very low bitrates.

Add/Remove Package

11. If required, click **Add Package** to create a new package. The new package will be automatically numbered. If you decide to delete a package by clicking **Remove**, the packages will be renumbered accordingly.
12. Click **Next** to proceed to the next step.

Package Review

An expandable list of the packages created by the system appears. Each package contains four assets: Best, High, Medium and Low quality along with their corresponding file names and bitrates:



Package #1 name: CarIntro

FLV7 Assets

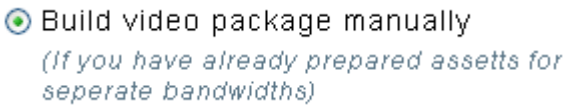
Type	File Name	Settings
Best Quality Asset	CarIntro_FLV7_Best	750 kbps
High Quality Asset	CarIntro_FLV7_High	500 kbps
Medium Quality Asset	CarIntro_FLV7_Medium	300 kbps
Low Quality Asset	CarIntro_FLV7_Low	100 kbps

FLV8 Assets

Type	File Name	Settings
Best Quality Asset	CarIntro_FLV8_Best	750 kbps
High Quality Asset	CarIntro_FLV8_High	500 kbps
Medium Quality Asset	CarIntro_FLV8_Medium	300 kbps
Low Quality Asset	CarIntro_FLV8_Low	100 kbps

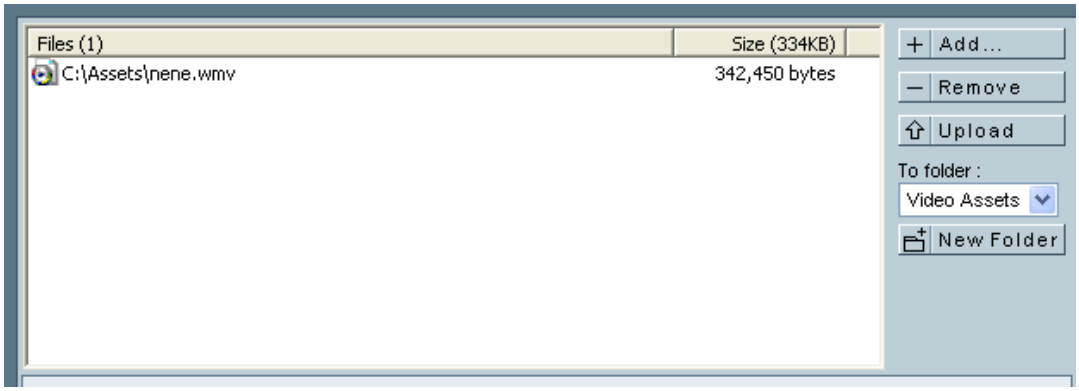
13. Click **Submit** to start uploading the files from your local disk and start the encoding process. An alert appears indicating the estimated time for completion of the package.
14. Click **OK** to close the notification window and continue the encoding process, or **Cancel Encoding** to stop the encoding process.

Track 2 - Build Video Package Manually

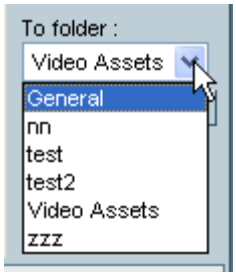


Upload (Skip this step if you have already uploaded the file on the creative assets page)

1. Click Upload to open the dialog for handling the assets you want to upload to the system.
2. Click **Add** and browse to the FLV or WMV you want to upload. You may select more than one file at a time. The selected file(s) will appear in the list.



3. From the **To folder** drop-down list select the folder in which you want to place the file.



4. Click **New Folder** and enter the name of a new folder if you need one.
5. Click **Upload** to upload them to the system.

   Note: It is not necessary to upload all the size categories in all the formats. The minimum you must upload is the highest bitrate for one format. But if you upload, for example, a medium bitrate FLV, you also have to also upload a High version of the FLV.

Package Selection

6. Enter a **package name**.

FLV 7 Assets (Optional)

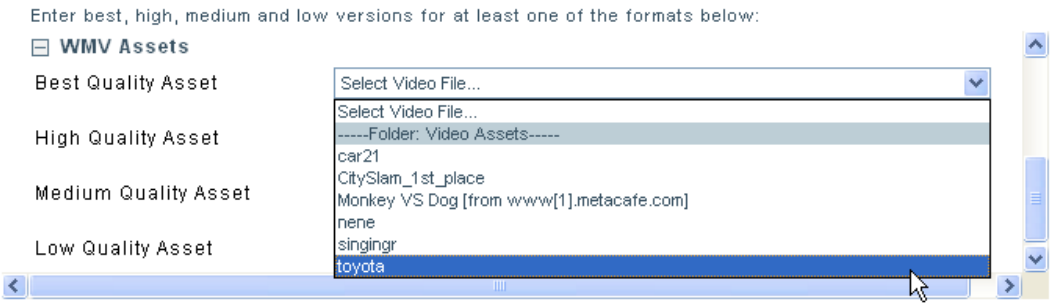
7. For each FLV 7 file size, select a file that was already uploaded to the system from the drop-down menu.

FLV 8 Assets (Optional)

8. For each FLV 8 file size, select a file that was already uploaded to the system from the drop-down menu.

WMV Assets (Optional)

9. For each WMV file size, select a file that was already uploaded to the system from the drop-down menu:



10. Enter free text in the **Notes** field which will help you to recognize the package later.
11. Click **Submit.** Your package is ready and you can preview it in the packages view of the Video Studio

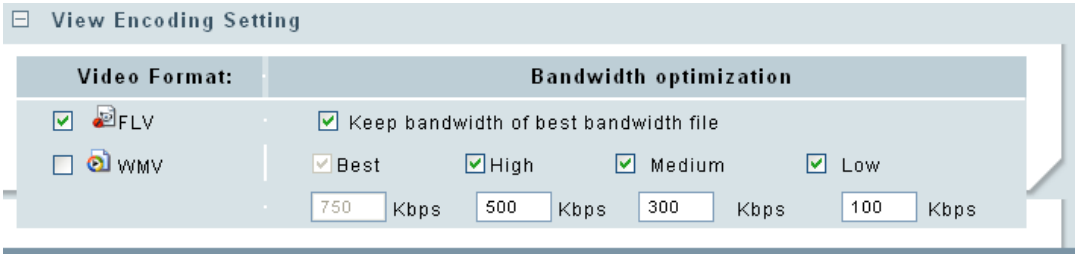
Track 3 - Upload raw video asset(s)

Package Selection

1. Enter a package name.
2. Enter free text in the **Notes** field which will help you to recognize the package later.
3. Enter **Special Instructions.** These are instructions for Eyeblaster services. For example, "Make the video optimized for movement".

View Encoding Settings (Optional)

Expand the **View Encoding Settings** section if you want to change the default settings. These default settings are the recommended ones:

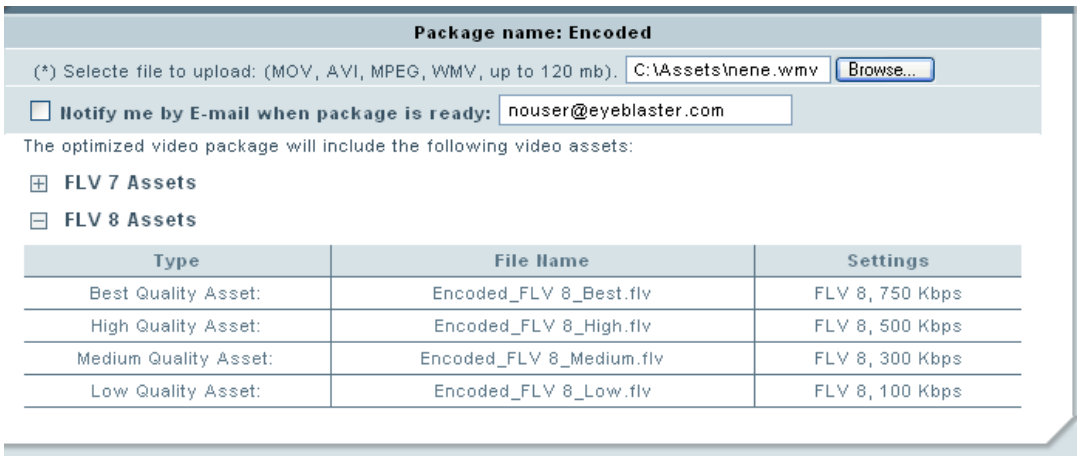


4. Check the checkbox next to the video formats you wish to create. (FLV, WMV or both).
5. Check the **Bandwidth** checkboxes and, if you want, change the default bitrate for **High**, **Medium** and **Low** bandwidths. Note that the default bitrates are based on the statistics from actual user bandwidths and are therefore the recommended values.

Upload and Review

6. Click **Browse** and select the file to be uploaded.

   An expandable list of the packages to be created by the system appears. The package contains a total of 12 assets: Best, High, Medium and Low quality along with their corresponding file names and bitrates:



Type	File Name	Settings
Best Quality Asset:	Encoded_FLV 8_Best.flv	FLV 8, 750 Kbps
High Quality Asset:	Encoded_FLV 8_High.flv	FLV 8, 500 Kbps
Medium Quality Asset:	Encoded_FLV 8_Medium.flv	FLV 8, 300 Kbps
Low Quality Asset:	Encoded_FLV 8_Low.flv	FLV 8, 100 Kbps

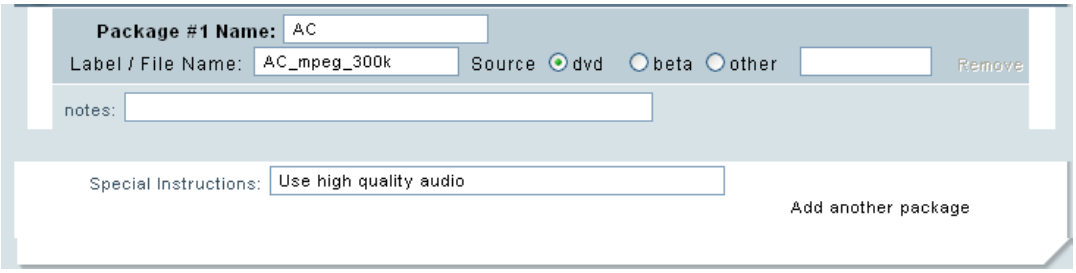
7. If you wish to receive notification by email when the package creation process is complete, mark the **Notify me by email when files are ready** checkbox.
8. Click **Submit** to start uploading the files from your local disk and start the encoding process. A progress bar is displayed indicating what percentage of the uploading is complete. (Do not close the progress bar window until the asset has finished uploading or this will stop the uploading process).
9. Click **OK** to close the notification window and continue the encoding process, or **Cancel Encoding** to stop the encoding process.

Track 4 - Send via Regular Mail to Eyeblaster

Package Selection

1. Enter a **Package Name.**
2. Enter a **Label/File name** . The recommended form is **<filename>_<format>_<bitrate>.** If you are sending multiple items, this label should make it easy for Eyeblaster Services to identify which media is for which package. For example, in the case of a CD, this name can also be what you defined as the label name when writing the CD.
3. Check the radio button next to the data storage item you intend to mail. If your media type does not appear, check **Other,** and specify what it is.
4. Enter free text in the **Notes** field which will help you to recognize the package later.

5. Enter **Special Instructions.** These are instructions for the service. For example, "Produce the video audio at highest quality"

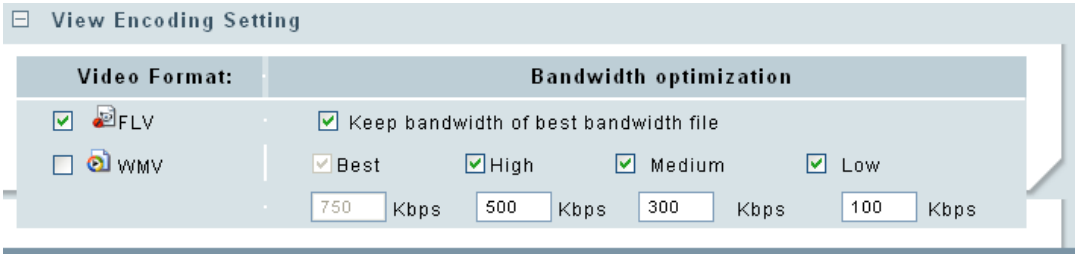


Add/Remove Package

6. If required, click **Add another package** to create a new package. The new package will be automatically numbered. If you decide to delete a package by clicking **Remove**, the packages will be renumbered accordingly.
7. Click **Next** to proceed to the next step.

View Encoding Settings

Expand the **View Encoding Settings** section if you want to change the default settings. These default settings are the recommended ones:



8. Check the checkbox next to the video formats you wish to create. (FLV, WMV or both).
9. Check the **Bandwidth** checkboxes and, if you want, change the default bitrate for **High**, **Medium** and **Low** bandwidths. Note that the default bitrates are based on the statistics from actual user bandwidths and are therefore the recommended values.

Package Review

The package slip details are displayed. This includes the Sender information taken from your account details as well as the package details.

10. To edit the package details click **Previous**.
11. If you wish to receive notification by email when the package creation process is complete, mark the **Notify me by email when files are ready** checkbox.
12. Click **Print** to produce a shipping label. This shipping label must then be attached to the storage items you wish to mail.
13. Click **Submit** to confirm the process was completed successfully.

Working in Flash

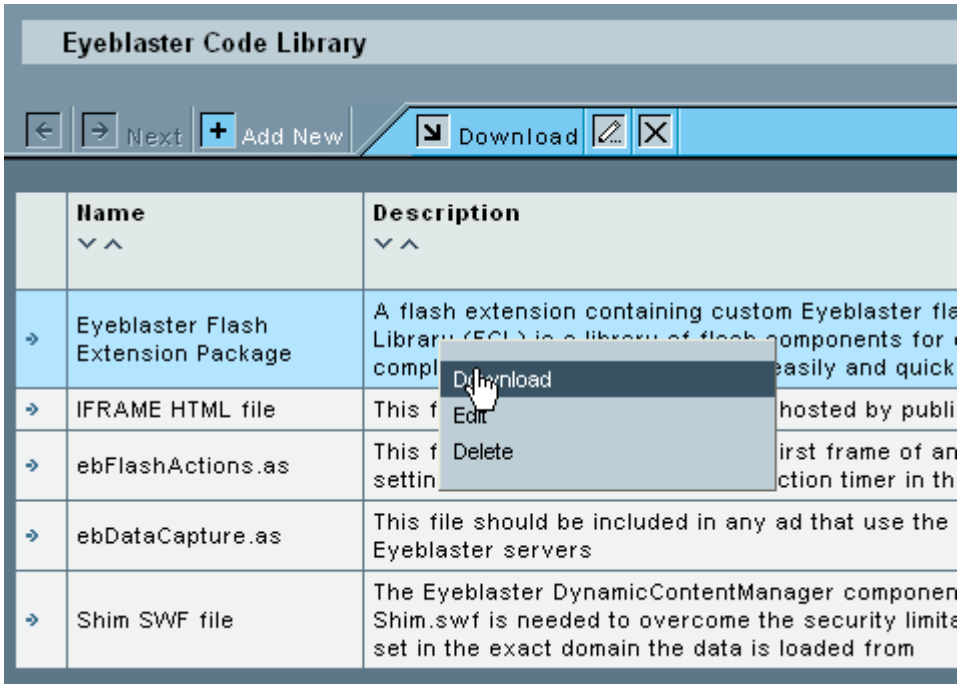
MXP Installation

To download and install the Eyeblaster *Flash* Extension Package:

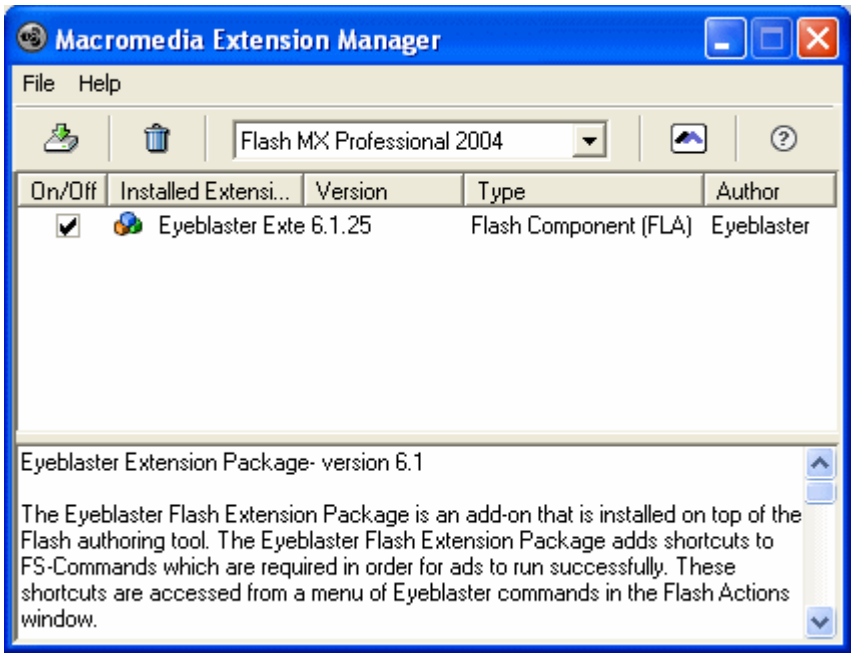
1. Browse to http://www.macromedia.com/exchange/em_download/, and install the Macromedia Flash Extension Manager. This is required before you can install Flash Extensions onto your Flash authoring software.

   From the Eyeblaster homepage, download the **Eyeblaster Flash Extension Package** by clicking on the link below:

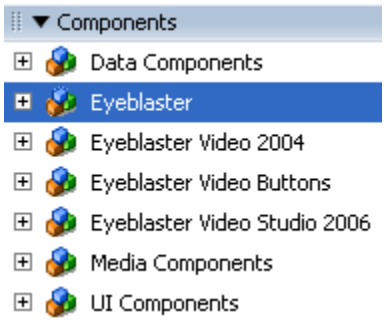
Alternatively, you can download it by going to the **Eyeblaster Main Menu**, clicking **Resources / Download Area / Code Library** and right-clicking **Eyeblaster Flash Extension Package** and clicking **Download** on the popup menu:



2. Double-click the downloaded file to install the extension. The Eyeblaster Flash Extension Package is now added to the Macromedia Extension Manager:



3. Open Flash and then open the **Components** tab. You will see the new groups of Eyeblaster components:



4. Right-click any of the commands to access the "Reference" link. This link will open the command reference, along with instructions and examples.

Video Components

The Eyeblaster MXP components are arranged as follows in the Flash Designer:

- **Eyeblaster** – all general EB components: Creative State, Dynamic Content Manager, Dynamic Mask, EBAdMEssaging.
- **Eyeblaster Video Studio 2006** – new video components: Video Loader, Video Controls, Video Playback
- **Eyeblaster Video 2004** – old video components: SWF Loader, FLVProgressiveLoader, FLVStreamingPlayer
- **Eyeblaster Video Buttons** – new video controls as separate components: Play/Pause, Stop, Mute, Replay, Seek Forward, Seek Backward, Playback Slider (with progress), Status Change, Volume

How to Create a Full Screen Skin

The following are a number of guidelines for creating a skin to be used for full screen viewing of video. This is when you choose to design your own full screen skin as opposed to selecting Eyeblaster's default full screen skin (EBFullScreen) when setting up the ad on the Eyeblaster Platform:

- It is recommended to place the Video Loader on a Flash stage that represents the size of the user's screen (e.g. 1024 X 768), and then to drag over the buttons. The Video Loader component should be made as large as possible. This is to avoid distorting the button size which occurs if you place the buttons on a smaller stage.

  Note that the customized full screen skin disregards the parameters defined in the Video Loader interface, e.g. the settings of the **Online video name** as it only enlarges the current movie. It also disregards the sound options as sound options for full screen are defined externally through the Eyeblaster platform during ad creation.

- It is good practice to include a full screen close button  (XBtn component) which can be positioned anywhere on the stage.

Full Screen Video Implementation

There are a number of ways to implement full screen video playback. These methods are a solution for providing different user experiences depending on publisher restrictions and bandwidth considerations.

- Single video asset

  The first method is to create one package for both regular viewing and full screen viewing of the ad. This method provides a seamless transition from regular plackback to full screen viewing upon user initiation (the video continues playing but in full screen mode) but involves the initial uploading of a heavy asset which may conflict with publisher restrictions.

  To use this method, you need to select the same value (ebMovie1 - ebMovie6) in the “Main Ad Video” and “Full Screen Video” fields in the Video Loader or Video Playback components. You also need to create a single package using the VideoStudio and place it in the additional asset tab with the correct ordinal number. For example, if you selected "ebMovie1" you should place the package in additional asset ordinal #1.

- Two video assets

  The second method is to create two separate assets one for regular payback and the second for full screen viewing. This enables you to reduce the size of the asset loaded during initial loading in compliance with publisher restrictions and to take advantage of a high kpbs in a separate asset for full screen viewing which is only loaded upon user initiation. This is achieved by selecting the separate asset in the Full screen video field in the Video Loader or Video Playback components. The disadvantage is that the movie will start from the beginning when the user initiates full screen viewing. Note that the total size of both the regular playback package and the full screen package must not exceed 4.4 Mb.

- Full Screen Loader

  The third method allows you to create a banner ad with the with a full screen button with the sole purpose of initiating the full screen viewing of a video. An example would be simple banner ad asking the user to click on it to view a trailer of a movie. The movie is then played in full screen mode. To implement this option, Eyeblaster provides the Full Screen Loader component which can be placed directly on the ad in the Flash Designer. Use the OpenFullScreen API method to define the playback mode.

  The advantage of this option is that the full screen asset is only loaded upon user activation.

New Video Features

A number of changes and enhancements have been made which affect the implementation of video in version 6.1 of the Eyeblaster Platform with the introduction of the new Flash component (Eyeblaster Video Studio 2006) which combines the functionality of the old SWFVideoLoader, FLVProgressiveLoader and FLVStreamingPlayer components (Eyeblaster Video 2004). The new component provides the ability to control the display and runtime properties of the video through a user-friendly interface and provides a complete video optimization solution for multiple users with different bandwidths.

To facilitate the new functionality, you need to familiarize yourself with a number of new features that have been introduced:

- **Full Screen Video**

  Full screen video is supported for all of Eyeblaster ad formats.

- **Automatic Bandwidth detection and serving**

  Automatic setup for video in multiple bitrates and formats

- **Unified components for progressive & streaming**

  The new Video component unifies the functionality of the FLVProgressiveLoader and FLVStreamingPlayer, video components.

- **Component Inspector**

  Eyeblaster's Component Inspector provides a straight-forward graphic user interface from which you can define the settings of the new Eyeblaster Flash components. It provides full control over the video display and runtime settings. Click here for details.

- **WMV Support**

  You can now include WMV files in your ad in addition to WMV's high quality and superior bandwidth detection.

- **Streaming and Progressive supported in one video**

  The new video APIs have been combined to support both streaming and progressive download. Whereas before you had to select the appropriate video component and drag it to the stage of your Flash document, the new unified component lets you select the required video delivery features and define all the user experience settings within the video loader. For a detailed description of the APIs click here.

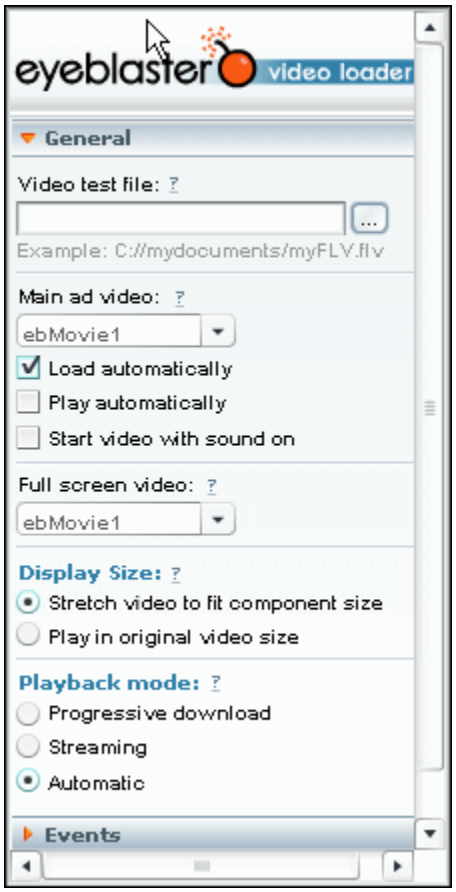
Eyeblaster Video Studio 2006 Components

Eyeblaster now offers an entirely new dimension in creativity and control over the video ad experience. With our new video loader component, creating high quality Video ads has become simple and intuitive. The video loader has an easy-to-use component inspector allowing you to customize the video experience using a few simple selections (instead of writing code), which together with the ability to customize and create skins, has simplified the creation of Video Ads into an extremely scalable process.

Eyeblaster Video Studio 2006 Components can be modified using the Flash Component Inspector giving you full control over the video:

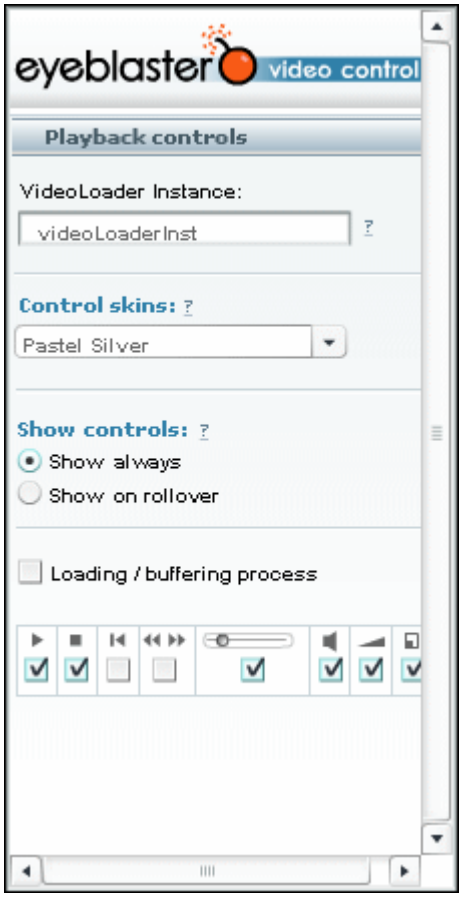
- **Video Loader**

  This component lets you control the various video settings for different kinds of video formats. This component should be used if you need to place the controls in different locations, or if there are no controls at all. It can be used alone or together with the **Video Controls** component.



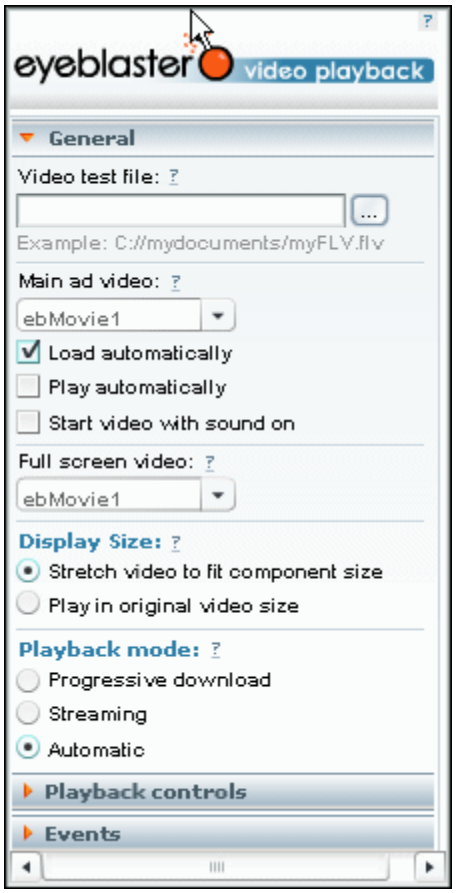
- **Video Controls**

  This component includes a user interface that lets you define the user experience of the video by defining which control buttons will appear on the video control panel. Each control button functionality is according to the API associated with it. Groups of buttons are divided into skins.



- **Video Playback**

  This is the combined video component which lets you control both the video settings and control buttons. It is recommended to use this component unless you wish to place the control panel somewhere else other than below the video play area.



- **Full Screen Loader**

  This component, which can be placed on the ad, opens the movie directly in full screen mode.

  It contains 2 parameters:

  - **ebMovie 1-6**. The additional asset to be used specially for full screen play. Note that the full screen controls and skin are defined in the default EBFullScreen skin selected in the Eyeblaster Platform when setting up the video ad while the sound settings are according to the ad setup settings It is recommended to select **sound ON** or **sound OFF** in the ad setup as apposed to **maintain current user setting** which uses the settings in the EB full screen skin.
  - **Playback Mode**: Defines how the movie is played.

Editing the Video Components

This procedure involves preparing the video *skin SWF* (container) inside the Flash Designer by defining the various display and runtime options. The skin can be an entire Flash experience that encompasses the video area. Eyeblaster provides a variety of default skins which can be accessed from the VideoControls and VideoPlayback components or individual control buttons (Eyeblaster Video buttons) which can be customized and edited.

***Note**: When adding an include statement (ebFlashActions or ebDataCapture) when using "script assist" in the Flash 8 authoring tool or using export mode in Flash MX, it automatically ads a semicolon at the end of the include statement resulting in a script error. Delete the semicolon to fix this error. (Note that this error does not occur in expert mode in Flash MX, or the default mode in Flash 8).*

Video Loader Options

The following options can be defined in the component interface:

Video Loader - General Settings

- **Video test file**: This option allows you to test the video ad without the need to upload the video first to the Eyeblaster platform. The offline testing source is actually the video file (FLV) on your local drive. This is also the asset which you will later upload for package creation or as an additional asset in the Eyeblaster Platform. It is only possible to test an FLV using this option. If you would like to test a WMV video, the video asset must be first uploaded to the Eyeblaster platform and placed in the additional assets folder of the ad you are testing. For more information, refer to the video ad setup step.
- **Online video name**: Once testing is done, the video file has to be uploaded to the Eyeblaster platform for runtime playback. It is possible to use the video file as-is, or create a video package from it. To learn more about the benefits of video packages and how to create them, click here). The option in this list box corresponds to the ordinal number that the additional asset will be given when loaded into the Eyeblaster platform. For example, ebMovie1 corresponds to the first additional asset, ebMovie2 to the additional asset appearing with the numeral 2 and so on.
- **Load Automatically:** As soon as the Flash asset is loaded, the video asset is loaded. If this checkbox is not checked, you need to programmatically load the video in the actionscript by using the ***VideoLoad*** API.

Description	Used to download a *Flash* movie containing a video, and start playing the video, automatically when ready.
Usage	`videoLoad(ebMovie<number>, nFrameRate, nProgressiveDownloadRisk);`

Parameters	" ebMovie<number> [string] The ordinal number of the video as indicated on the Eyeblaster Interface under Additional Assets. For instance, for video number 2, you would enter: ebMovie2. There is no default for this parameter. " nFrameRate [integer] *Frame rate* of the video. There is no default for this parameter. This parameter is used to calculate the *buffer* size of the progressive download. " nProgressiveDownloadRisk [float] This parameter is used to indicate the amount of additional video to download before starting to play, after the video indicates that it is buffered and ready to begin playing. Enter a number between 0 and 1, where 0 indicates no additional downloading, and 1 indicates that the buffer size will double itself. There is no default for this parameter, but the recommended value is 0.1.

Return Value	None.
Additional Information	Usually a video will indicate that it is ready to begin playing when the time required to download the remainder of the video is less than the length of the video. By increasing nProgressiveDownloadRisk, you force additional downloading before the video begins playing. Use videoLoad to preload a video that should only start playing after a *user* interaction, such as pressing a Start button. Use videoLoadAndPlay to download and play a video without any user interaction.
Requirements	Flash 6.

Sample Code	`//load ebMovie1, use frame rate of 15 fps (the fps of the movie we use)`
	`//force additional downloading of 10% before the video begins playing.`
	`myswf_loader.videoLoad(ebMovie1, 15, 0.1);`

- **Play Automatically**: As soon as the Flash asset is loaded, the video asset plays (auto-initiated). If this checkbox is not checked, the user will have to click the Play button to start playing the video (user-initiated).

Mute Options

- **Start Video with Sound On:** Start playing the video unmuted (does not affect full screen viewing settings). If the checkbox is unchecked, the video will start muted.

Display Size

- **Best Fit to Component Size**: The movie is resized to best fit the play area. The movie is enlarged or reduced while maintaining its aspect ratio. If it is reduced it will be centered accordingly.
- **Play in Original Movie Size**: If the movie area is larger than the component area it will be reduced to fit. If the movie area is smaller than the component area it will play as is.

Playback Mode

The playback mode controls if the video plays using streaming or progressive download. Since some publishers require that the playback to be streamed, it is recommended to select the **Automatic** option. This ensures that the video will play according to the Publisher's requirements or in the most optimized manner based on Eyeblaster's advanced algorithms. Furthermore, if you have a campaign that requires that the same video asset be delivered as streaming for some publishers, and in progressive download mode for others, selecting **Automatic** saves you having to create separate assets for streaming and progressive. The trafficker can then select the appropriate setting on the Eyeblaster Platform. See Delivery Considerations below.

- **Progressive Download Only**: Deliver the video in progressive download mode.
- **Streaming Only**: Stream the video.
- **Automatic**: Deliver the video according to the settings on the Eyeblaster Platform.

Video Loader - Events

On Click

These are the events that will occur when the user clicks on the video play area.

- **Go to Default Clickthrough URL**: A new window opens to the URL defined in the Eyeblaster platform.
- **Play/Pause**. The movie toggles between play and pause each time the user clicks.
- **Custom Behavior**: The function for `<instanceName>onClick` within the Flash actionscript is called. Click the **Eyeblaster** layer in the Flash timeline in order to edit the code.
- **None**: Nothing occurs when the user clicks.

On Roll Over:

These are the events that will occur when the user moves the mouse cursor over the video play area.

- **Mute/Unmute**: The movie toggles between Unmuted when the user moves his mouse cursor over the play area, and Muted when he/she leaves the play area.
- **Play/Pause**. The movie toggles between Play when the user moves his mouse cursor over the play area, and Pause when he/she leaves the play area.
- **Custom Behavior**: The function for `<instanceName>onRollover` within the Flash actionscript is called. Click the **Eyeblaster** layer in the Flash timeline in order to edit the code.
- **None**: Nothing occurs when the user moves the mouse over the video play area.

On Movie End:

These are the events that will occur when the movie reaches the end of play.

- **Replay**: The movie restarts automatically at the end of play (looping). Auto-Initiated Replay, i.e. looping that is not user-initiated is not recommended in **Streaming** mode as it requires additional bandwidth usage each time the movie is played. To prevent this from occurring, this setting will automatically only replay the video once for streaming assets.
- **Close**: The movie automatically closes when finished playing.
- **Custom Behavior**: The function for `<instanceName>onMovieEnd` within the Flash actionscript is called. Click the **Eyeblaster** layer in the Flash timeline in order to edit the code.
- **None**: Nothing occurs automatically when the movie ends.
- **Go to Frame:** Once the video ends, the Flash timeline will continue from the selected frame.

Advanced Event Handlers:

These are advanced events provided by Eyeblaster that occur within the video loader and video playback component..

- **OnStatusChanged**: Every time the playback status changes, e.g. changing from *Play* to *Stop*, or *Play* to *Pause*, etc. the defined Flash function is called. As an example, a status bar can be implemented using this option. Click the **Browse** button to access the code with the user-defined parameters for this callback function in the first frame of the Flash asset. Click here for event details.
- **OnPlayProgress**: An event that records the movie progress. This allows a Flash function to be called at a certain point in the movie allowing further user interaction. As an example, the user will be able to do a mouse roll over on a element in the movie (for example when a certain character enters the frame) which will result in the user being prompted to click it and be redirected to another URL. Click the Browse button to access the code with the user-defined parameters for this callback function in the first frame of the Flash asset. Click here for event details.
- **OnBufferLoaded**: In Streaming mode, once the buffer is loaded a Flash function can be called. This gives you greater control over the streaming process. Click the Browse button to access the code with the user-defined parameters for this callback function in the first frame of the Flash asset. Click here for event details.
- **OnBufferProgress:** In Streaming mode, this event lets you call a Flash function while the buffer is loading. Click the Browse button to access the code with the user-defined parameters for this callback function in the first frame of the Flash asset. Click here for event details.
- **OnLoadProgress**: In Progressive Download mode, this event lets you call a Flash function while the buffer is loading. Click the Browse button to access the code with the user-defined parameters for this callback function in the first frame of the Flash asset. Click here for event details.
- **onError**: If an error occurs when trying to connect to the FCS or when trying to load and play a video that cannot be found, the defined Flash function is called. As an example, a message describing the error can be displayed using this option or a different video can be loaded/played. Click the Browse button to access the code with the user-defined parameters for this callback function in the first frame of the Flash asset. Click here for event details. Click here for event details

Video Controller

The Video Controller lets you select a control skin and define which controls will appear in the panel. Here you define the controls skin for full screen view mode and regular mode. You can use the Eyeblaster default full screen skin regardless of what skin you use for normal play. The full screen view skin typically contains fewer control buttons so as to give the user maximum view area of the video.

Control Skins: Select the skin to be used as the control panel from the drop-down list. The style of the skin is previewed below and you can define which buttons you want to include by checking/unchecking them accordingly. All changes made inside the **Component Inspector** are visible on the stage.

- **Play/Pause**: Includes a toggle **Play/Pause** button.
- **Stop**: Includes a **Stop** button. After the *Stop* button is clicked the movie will play from the beginning when the *Play* button is clicked.
- **Replay**: Includes a **Replay** button. This button plays the movie from the beginning (equivalent to *Stop* and *Play*). See Replay mode recommendations above.
- **Seek**: Includes a **Seek** button. This allows the user to "jump" a few seconds backwards and forwards within the movie.

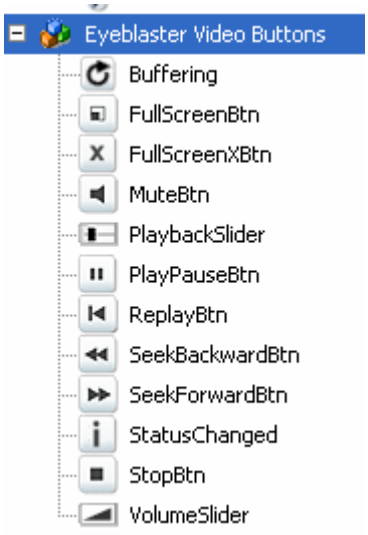
- **Slider**: Includes a slide bar. This allows the user to move backwards and forwards within the movie.
- **Mute**: Includes an **Mute/Unmute** toggle button.
- **Volume**: Includes a volume control bar.
- **Full Screen**: Includes a button that allows the user to toggle between full screen and regular view modes.
- **Loading/Buffering Progress**: Includes a control which shows how much of the buffer has been filled. Note that the **Playback Slider** (which lets you move back and forth to different points in the video) is also part of this component. Therefore, if you want to remove this element from your skin, you need to uncheck both the **Playback Slider** and the **Loading/buffering progress** checkboxes.
- **Playback Status**: Includes the following text message that indicates the playback status:
    - **Idle**: Before download starts.
    - **Loading**: Progressive download loading buffering (for streaming only).
    - **Ready**: Movie finished loading and can start playing (when "automatically play" is not used).
    - **Playing**: Movie is playing.
    - **Paused**: Movie is paused (play will continue from same spot).
    - **Stopped**: Movie stopped (play will restart movie).
    - **Full Screen Playback**: Full screen is playing in another window.

Show controls

- ***Always Show:*** The control panel is fixed and always visible.
- ***Show on Rollover:*** The control panel only appears when the user does mouse rollover. If the mouse remains stationary, the control panel will disappear after a few seconds.

Editing Eyeblaster Control Buttons

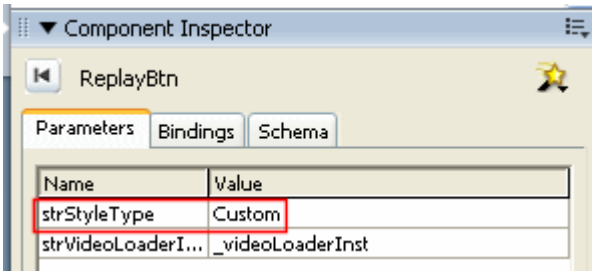
The individual control buttons within the **Eyeblaster Video Buttons** directory below can edited either on the stage or in the library.



- **Editing on the Stage**

  This option uses the "Custom" style and allows you to customize the style.

  1. Ensure that under **Properties** in the **Parameters** tab, the style **Custom** for **strStyleType** is selected. (This is usually the default selection).



  2. Edit the component on the stage. Note that only the editable layer is visible and not locked.

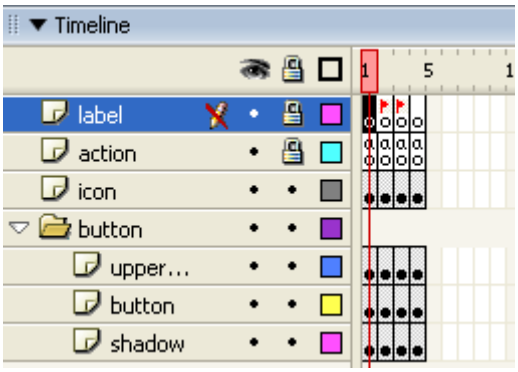
  This editable layer contains a movie clip called "_icon_mc" with the following elements:

  - 1st frame: a button icon (As it appears on the stage in design mode) - **this must not be changed**.
  - A frame with a matching label for each of the button states.

  The various buttons have different editable states. For example, a toggle button such as **Play/Pause** may contain 2 or 3 editable states whereas a regular button, such as **Stop** may only have 1 editable state.

Regular Buttons

For the regular buttons, "`_icon_mc`" contains 4 frames:



- 1st frame - the button icon (as it appears on the stage in design mode).
- 2nd frame - the start (init) state of the button, e.g. `ebINIT_Replay.`
- 3rd and 4th frames - the transition between states, eg `ebREPLAY.`

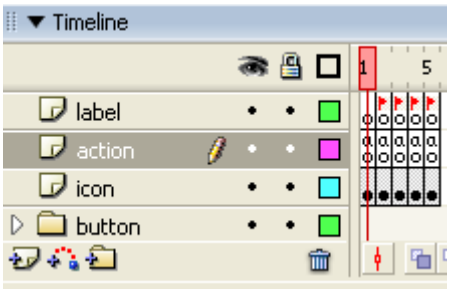
The movement (transition) starts in the 3rd frame and ends in the 4th frame.

The 3rd frame should contain the starting (init) state of the movement and must be with no movement (usually the same as the 2nd frame)

The 4th frame should contain the movement.

Toggle Buttons

For the toggle buttons, "`_icon_mc`" contains the following frames:

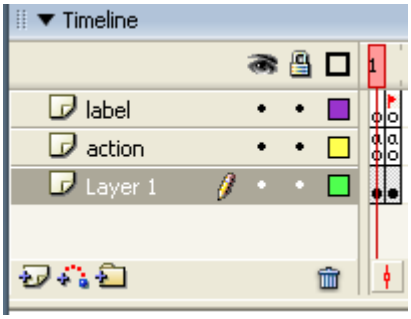


- 1st frame - the button icon (as it appears on the stage in design mode).
- A frame for each starting (init) state, e.g. `ebINIT_Mute, ebINIT_UNMUTE.`
- A frame for each transition used to display animation when the button is pressed, e.g. `ebMUTE, ebUNMUTE.`

Note that the state in the movie clip refers to the end state and not the starting state. In other words, for Mute, the `ebMUTE` represents a transition from an "Unmute" state to a "Mute" state.

Slider Button

For the slider button (`VolumeSlider` and `PlaybackSlider`), _icon_mc contains following frames:



- 1st frame - the button icon (as it appears on the stage in design mode).
- 2nd frame (SLIDER) - the slider itself.

The slider consists of 4 movie clips:

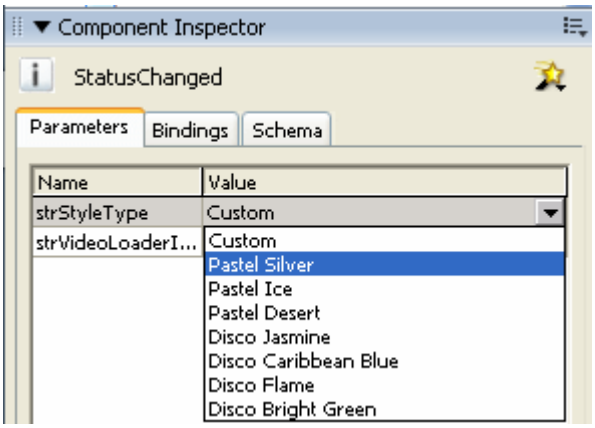
- `_slider_mc` – the slider moving head
- `_slider_line_mc` – the line the slider moves on
- `_bar_mc` – the slider bar
- `_progress_bar_mc` (PlaybackSlider only) – the loading/buffering progress bar.

Note: You can edit or create new movie clips to replace each of the slider movie clips **but you must not change their name.**

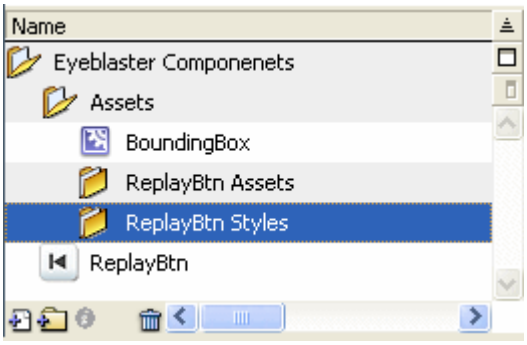
- **Editing in the Library**

  This option entails editing a predefined style.

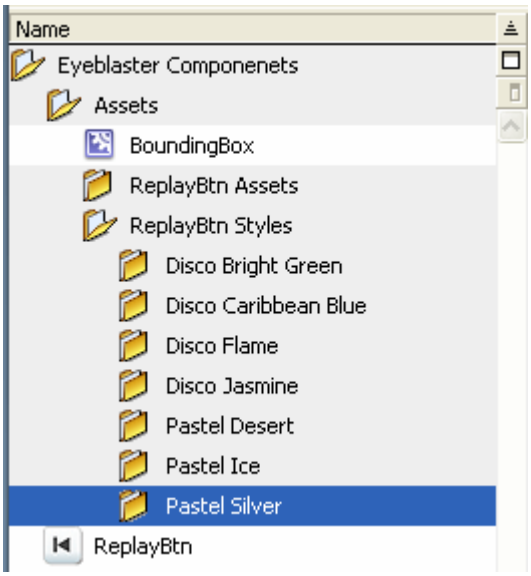
  1. Under **Properties** in the **Parameters** tab, select the required style, e.g. **Pastel Silver**, for `strStyleType`:



2. In the **Library** under **Eyeblaster Components**, (press F11 to view if not already displayed), open the **Assets** folder and go to the folder **<ButtonName> Styles**.

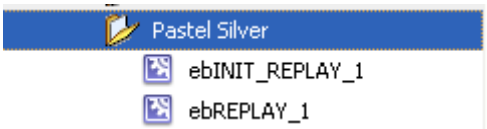


Within this folder are subfolders containing the editable button graphics. For example, if you want to modify the "Pastel Silver" style you selected for the Mute Button, you need to go to the **MuteBtn Styles** folder and open the **Pastel Silver** folder and edit the styles in this folder:

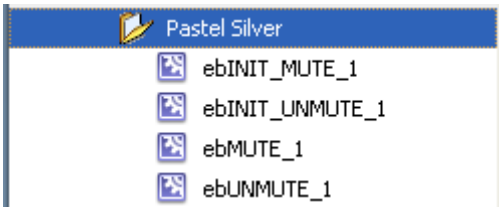


Note that the various buttons have different editable states. For example, a toggle button like *play/pause* may contain 2 or 3 editable states whereas a regular button, like *Stop*, may only have one editable state.

The style folders for the regular buttons contain 2 movie clips (one for each editable state): the initial movie clip for the starting state of the button, e.g. `ebINIT_Replay` and the transition movie clip for the transition between states, e.g. `ebREPLAY:`



The style folders for the toggle button contain an initial movie clip for each state, e.g. `ebINIT_Mute, ebINIT_UNMUTE` as well as a transition movie clip for each transition used to display animation when the button is pressed, e.g. `ebMUTE, ebUNMUTE:`



Note that the state in the movie clip refers to the end state and not the starting state. In other words, for Mute, the `ebMUTE` represents a transition from an Unmute state to a Mute state.

Defining the Loading Animation

Eyeblaster provides a loading animation component which lets you define what Flash animation will be played while the video is loading. This is done by dragging the Buffering component  in the **Eyeblaster Video buttons** folder to the stage and editing the enabled layer. You can then delete it from the stage and the selected Flash video will play when the video is loading.

How to test the video file

The following guidelines show you how to test the various video formats within the Flash Designer.

If you wish to test an FLV file, you can simply go to the **Video test file** field in the Video Loader component interface and select it directly from your hard disk.

A WMV cannot be tested from within the Flash Designer. However, once the SWF and the WMV file(s) are loaded into the Eyeblaster Interface, they can be tested there.

Eyeblaster Video Studio 2006 API

The new Video component unifies the functionality of the FLVProgressiveLoader and FLVStreamingPlayer, video components into one new Video Loader component. Whereas before you had to select the appropriate video component and drag it to the stage of your Flash document, the new unified component lets you select the required video delivery features and define all the user experience settings within the video loader. Nevertheless, you still have the option to modify video APIs within the Flash document, if required.

The following is a list of the new video API methods that should be used only if the API Eyeblaster provides needs to be customized. Note that for the most part it is not necessary to modify the methods except in special cases.

<table>
<tr><th rowspan="2">Method</th><th colspan="3">Old Method that was Replaced</th></tr>
<tr><th>FLV Streaming</th><th>FLV Progressive Download</th><th>SWF Video</th></tr>
<tr><td>VideoLoad (nMovieNum,nFSMovieNum)
<table>
<tr><td>Description</td><td>Used to download a video file.</td></tr>
<tr><td>Usage</td><td><code>VideoLoad(nMovieNum, nFSMovieNum);</code></td></tr>
<tr><td>Parameters</td><td>" <code>nMovieNum</code> <number> [string]"
The ordinal number of the regular view video as indicated on the Eyeblaster Interface under Additional Assets. For instance, for video number 2, you would enter: ebMovie2. There is no default for this parameter.
<code>nFSMovieNum</code> <number> [string]"
The ordinal number of the full screen video as indicated on the Eyeblaster Interface under Additional Assets. For instance, for video number 2, you would enter: ebMovie2. The default value is nMovieNum.</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
<tr><td>Additional Information</td><td>None</td></tr>
<tr><td>Requirements</td><td>Flash 7.</td></tr>
<tr><td>Sample Code</td><td><code>myVideo.VideoLoad(1,2);</code></td></tr>
</table></td><td>Init</td><td>Init + Load</td><td>Load (FrameRate - ?)</td></tr>
<tr><td>VideoLoadAndPlay (nMovieNum,nFSMovieNum)
<table>
<tr><td>Description</td><td>Used to download and play a video file.</td></tr>
</table></td><td>Init + SetAutoPlay + Load</td><td>Init + LoadAndPlay</td><td>LoadAndPlay</td></tr>
</table>

Usage	`VideoLoadAndPlay(nMovieNum, nFSMovieNum);`
Parameters	`nMovieNum`:Number - the video ordinal number, for example 1 for "ebMovie1" `nFSMovieNum` Number - the video ordinal number of the full screen video asset.
Return Value	None.
Additional Information	None.
Requirements	Flash 7.
Sample Code	`myVideo.VideoLoadAndPlay();`

`VideoPlay()` *`VideoPause()`*	Play Pause Stop	Play Pause Stop	Play Pause Stop

`VideoPlay()`

Description	Used to start the video playback.
Usage	`VideoPlay();`
Parameters	None.
Return Value	None.
Additional Information	This function is usually invoked as the result of a user trigger, such as releasing a button, or moving the mouse.
Requirements	Flash 6.
Sample Code	`myVideo.VideoPlay();`

`VideoPause()`

Description	Used to pause the video playback.
Usage	`VideoPause();`
Parameters	None.
Return Value	None.
Additional Information	This function is usually invoked as the result of a user trigger, such as releasing a button, or moving the mouse.
Requirements	Flash 6.
Sample Code	`myVideo.VideoPause();`

VideoStop()

Description	Used to stop the video playback.
Usage	`VideoStop();`
Parameters	None.
Return Value	None.
Additional Information	This function is usually invoked as the result of a user trigger, such as releasing a button, or moving the mouse.
Requirements	Flash 6.
Sample Code	`myVideo.VideoStop();`

VideoSetMute (ebMuteVal)

SetMute

N/A

Mute

Description	Used to set or unset speaker mute.
Usage	`VideoSetMute(mute);`
Parameters	" mute [Boolean] If set to true, mute is enabled. If set to false, mute is disabled. There is no default for this parameter. The default value before you call this function is false.
Return Value	None.
Additional Information	The value that you set is the percentage of the current speaker volume as set by the operating system.
Requirements	Flash 7.
Sample Code	`myVideo.VideoSetMute(true);`

VideoGetMute

New

New

Description	Returns the mute state (1 when muted, 0 when unmuted).
Usage	`VideoGetMute();`
Parameters	
Return Value	None.
Additional Information	None.
Requirements	Flash 7.

	Sample Code: `myVideo.VideoGetMute();`			
VideoSetVolume(VolLevel) Description: Used to set the speaker *volume* for the video. Usage: `VideoSetVolume(nVolumeLevel);` Parameters: " nVolumeLevel [integer] Speaker volume as a percentage. There is no default for this parameter. The default value before you call this function is 100. Return Value: None. Additional Information: The value that you set is the percentage of the current speaker volume as set by the operating system. Requirements: Flash 6. Sample Code: `myVideo.VideoSetVolume(50);`		SetVolume	N/A	SetVolume
VideoGetVolume Description: Returns the volume level as a number (0-100). Usage: `VideoGetVolume();` Parameters: nMovieNum:Number - the video ordinal number, for example 1 for "ebMovie1" Return Value: None. Additional Information: None. Requirements: Flash 7. Sample Code: `myVideo.VideoGetVolume();`		New		New
VideoGetLength Description: Returns the length of the video, in seconds. Usage: `VideoGetLength();` Parameters: None. Return Value: An integer containing the length of the video, in seconds. Returns 1 if the information is not yet available		GetLength	New	New

	Additional Information: The value that you set is the percentage of the current speaker volume as set by the operating system. Requirements: Flash 7. Sample Code: `var length = myVideo.VideoGetLength();`			
VideoSeek(s) Description: Used to jump to a specific location in the video, as specified by seconds from the beginning of the video. Usage: `VideoSeek(TimeInSec);` Parameters: " TimeInSec [integer] Absolute location within the video, specified in seconds. Return Value: None. Additional Information: None. Requirements: Flash 6. Sample Code: `myVideo.VideoSeek(0);`		Seek	N/A	N/A
VideoSetBuffer (ebBuffer) Description: Used to set the buffer size for streaming or the risk level for progressive download. Usage: `VideoSetBuffer();` Parameters: `nBuffer`:Number - The buffer/risk size. The parameter can have one of the following constants or any other number between 0-100 for streaming and 0-1 for progressive download. • `PlayerConstants.ebHigh` for high bandwidth • `PlayerConstants.ebMed` for medium bandwidth • `PlayerConstants.ebLow` for low bandwidth Return Value: None. Additional Information: None.		SetBufferTime	SetRisk	3rd parameter of Load

	Requirements: Flash 7. Sample Code: `myVideo.VideoSetBuffer();`			
VideoGetBuffer Description: Returns buffer/risk size. Usage: `VideoGetBuffer();` Parameters: Return Value: None. Additional Information: None. Requirements: Flash 7. Sample Code: `myVideo.VideoGetBuffer();`		New	GetRisk	New
OpenFullScreen(nMovieNum,nPlaybackMode) Description: Used to open a full screen window with the selected ebMovie file and playback mode. Usage: `OpenFullScreen(nMovieNum, nPlaybackMode);` Parameters: `nMovieNum`:Number - the video ordinal number, for example "1" for "ebMovie1" `nPlaybackMode`:Number - indicates the Playback mode: • Progressive Download(0); • Streaming Only(1) • Automatic(2) Return Value: None. Additional Information: None. Requirements: Flash 7. Sample Code: `myVideo.OpenFullScreen(1,2);`		New	New	New
VideoSetFullScreen (ebFullScreenVal, ebFullScreenType) Description: Returns full screen/page mode. Usage: `VideoSetFullScreen();`		New	New	New

Parameters	
Return Value	None.
Additional Information	None.
Requirements	Flash 7.
Sample Code	`myVideo.VideoSetFullScreen();`

`VideoGetFullScreen`

New | New | New

Description	Returns full screen/page mode.
Usage	`VideoGetFullScreen();`
Parameters	
Return Value	None.
Additional Information	None.
Requirements	Flash 7.
Sample Code	`myVideo.VideoGetFullScreen();`

`VideoFSOpen`

New | New | New

Description	Sets video to play in fullScreen/page.
Usage	`VideoFSOpen();`
Parameters	
Return Value	None.
Additional Information	None.
Requirements	Flash 7.
Sample Code	`myVideo.VideoFSOpen();`

`VideoFSClose`

New | New | New

Description	Sets video back to regular screen size.
Usage	`VideoFSClose();`
Parameters	`nAutoClose`:Number - the screen status, values: • `ebFSClose` = 0 (default),

	• `ebFSAutoClose = 1`			
Return Value	None.			
Additional Information	None.			
Requirements	Flash 7.			
Sample Code	`myVideo.VideoFSClose();`			

`VideoGetStatus`

Description	Retrieves the current status of the video.	New	New	New
Usage	`VideoGetStatus();`			
Parameters	None			
Return Value	Statuses: • Idle • Loading • Buffering • Ready • Playing • Paused • Stopped • Full Screen Playback			
Additional Information	None.			
Requirements	Flash 7.			
Sample Code	`myVideo.VideoGetStatus();`			

Setting up the Ad on the Eyeblaster Platform

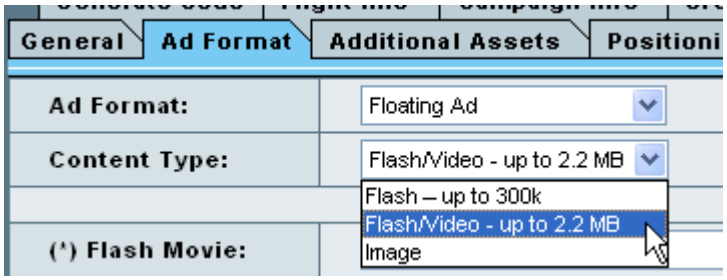
Once you have created the package, you need to attach it to the ad as follows:

In the **Edit Ad** dialog, select the required **Format Type** from the drop-down list.

To attach regular size video

1. In the **General** tab of the **Edit Flight** dialog, check the "**This flight includes video ads**" field and click **OK** to confirm that an additional fee is charged for including video:

2. In the **Video starts** field click the radio button to **Automatically** start playing the video or **By user interaction**.
3. In the **Ad Format** tab in the **Edit Ad** dialog of the ad for which you wish to add video, select **Flash/Video up to 2.2 MB:**



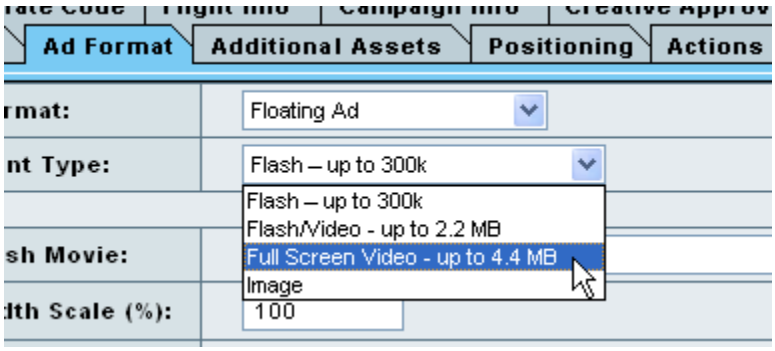
4. In the **Video starts** field click the radio button to **Automatically** start playing the video or **By user interaction**. Choosing **Automatically**, is dependent on this setting being selected at the flight level.
5. If you wish to place the video in the regular banner, then select the skin SWF as the **Default Flash**. Alternatively, if you wish to place the video in one or more panels, select the skin SWF as the **Panel Flash**. You can also place the skin SWF in both the regular banner and the panel.

Full Screen Video

1. In the **General** tab of the **Edit Flight** dialog, check the **This flight includes full screen video ads** checkbox. Note that the CPM for this feature may incur an additional fee which you need to confirm by clicking **OK** in the message dialog:

2. In the Video starts field click the radio button to **Automatically** start playing the video or **By user interaction**.

3. In the **Ad Format** tab in the **Edit Ad** dialog of the full screen video ad, select **Full Screen Video** from the **Content Type** drop-down list.



4. In the **Video starts** field click the radio button to **Automatically** start playing the video or **By user interaction**. Choosing **Automatically**, is dependent on this setting being selected at the flight level.
5. Select the skin SWF to be used as the **Full Screen Skin** i.e. which skin will be displayed when the video is viewed in full screen mode. Alternatively, you can select the default Eyeblaster full screen skin from the drop-down menu. In addition you need to define the **Full Screen Mute Options**.
    - **Start muted**: When the user views the movie in full screen mode, it will play muted regardless of what was defined in regular mode. (Default.
    - **Start unmuted:** Regardless of the regular view settings, the movie will play muted.
    - **Maintain current user setting**: When the user changes to full screen mode, the movie plays in whatever mode the user is currently in (muted or unmuted).

    Note that these full screen settings will be applied to all instances of full screen viewing in the ad, as in cases where you have multiple panels that can be viewed in full screen mode. This is regardless of whether or not you have different full screen skins for each panel. These mute settings also override the mute settings in the Flash swf.

Next you need to add the video package as an additional asset in the **Additional Assets** tab. "EBMovie1" will then access this package (defined as the first additional asset), and during the movie runtime, the most suitable asset within this package will be served. For download options see the Delivery Considerations section below.

Note:

- *When you select **This flight includes video ads** or **This flight includes full screen video**, the system automatically prevents the movie from being served to users with bandwidths of 56K or less (dial up). You can change this setting by checking the **Show on all bandwidths** in the bandwidth section of the **Filters** tab.*
- *In order to use the new video components and serve FLV or WMV video, the user is required to have Flash Player 7 installed.*
- *Video Studio does not support the SWF format in package creation. If SWF video is being used, the SWF Loader component should be used and the SWF video instead of a package is added as an additional asset.*

Delivery Considerations

The delivery method options defined in the Flash component are **Streaming**, **Progressive Download** and **Automatic**. The recommended choice is **Automatic**.

Video Enhancement Guidelines

The following is a list of guidelines pertaining to video creation:

- **Bandwidth considerations**

  In the automatic track for creating a package, the default movie encoding is 100, 300, 500 & 750 Kbps for low, medium, high and best quality assets respectively. The IAB recommends not showing video to users with connection speeds of less than 200 Kbps. However, since you can control the size of the display, you can ensure that even users with less than 200 Kbps can enjoy a positive video experience. Nevertheless, if in certain situations you wish to limit the video to users with connection speeds of 200 Kbps and above, the best method is to change the default bandwidth settings to 200, 350, 500 and 750, and also create a bandwidth filter to filter out 56, 100 and 200 Kbps in the ad setup process on the Filters tab.

  By default, every video ad is created with a filter that excludes dialup users (56 Kbps) consistent with Eyeblaster's policy of matching the video experience to the user's conditions. It is recommended to create a separate Flash Ad and serve it to dial-up users only by filtering out the higher bandwidth levels.

- **Interactive on-top layer**

  Since interactivity is not supported on top of a WMV, you will have to disable WMV if you create an interactive layer as part of your Flash resource. The easiest way is to not include the WMV format in the package creation process.

- **Customizing and saving a skin**

  The default Eyeblaster skin which comes with the video components in the MXP has various styles. In order to edit the graphics it is recommended to choose the “custom” style which allows you to edit the graphics directly on the stage. In order to save this skin, the file should be published as an swf and loaded to the RMP. If the component is opened in a new Flash document it will loose the editing changes, but it is possible to drag the component from the original document where it was edited into the new document.

- **Customizing the loading animation**

  The loading animation is a movie clip which can be edited. To reuse an edited animation, you can drag it into a new document.

- **Streaming vs. Progressive Download**

  The general consensus in the industry is that streaming is better than progressive download. However, this is only when “better” is defined as the ability to start playing the video as quickly as possible. If part of the consideration is also the experience during playback, streaming has some problems: in some cases it can cause the movie to stall while the streaming buffer is being filled. In progressive download, a long enough portion of the file is downloaded to ensure that the video will play without stalling. Eyeblaster improved the existing progressive support for FLV and offers an advanced progressive solution that minimizes the wait time before the video can start playing.

  In general, it is recommended to use streaming only for videos over 1 minute long. For more information, see the Streaming vs progressive download section.

- **Campaigns running on Yahoo! or in the Japanese market**

  Videos running on Yahoo! or in cases where they have to be run using streaming, it is recommended to use “Automatic” as the choice for the Playback option in the Flash component so that the same ad will run with progressive download not on Yahoo! and with streaming when running on Yahoo!

Advanced Topics

How does Plugin Detection Work?

The plugin detection tool checks which video player plugins the user has installed. It determines the preferred player in the following order: WMV, FLV 8 and FLV 7. For example, if you have all 3 formats in your package, the system looks for the WMV plugin (Media Player 9 and above), if this is not found it looks for FLV 8, if this is not found it looks for FLV 7. If none of these plugins are detected, no video is served to the user.

The next check the tools does is to determine the user's bandwidth and to serve the movie with the bitrate that best suits their bandwidth.

Content Delivery Methods

Progressive Downloading

In a progressive download, the content, or a portion of it, must be buffered (preloaded) to memory before a recipient can play it. The allowable length of any clip delivered is limited to the hard disk space and RAM available on a *user*'s machine. Generally progressive movies should not be made to last more than a few minutes. Content plays as soon as it is ready, so on fast enough connections, progressive movies can appear to be *streaming*. However, for a user to advance ahead in the clip, he or she must download everything from the point currently in view to the point of interest.

The main advantage of progressive downloads is its high quality at any connection speed. Given its superior quality, progressive video has proven very popular in the entertainment industry. It may take a lot longer for a user on a 28.8 kbps modem connection to download the same clip as someone with a much faster cable modem, but once downloaded, the clip will have exactly the same quality on each machine.

Progressive downloads are sent using the *Hypertext Transfer Protocol (HTTP)* and can be delivered using ordinary Web servers. HTTP is part of the connection-oriented protocol suite called Transmission Control Protocol/Internet Protocol (TCP/IP). Connection-oriented protocols guarantee the safe delivery of every packet sent. Since progressive downloads use the same protocol as common Web content, there is less of a chance of encountering problems getting past firewalls than with content from a streaming *server*.

For a table comparing progressive downloading with streaming, see the FAQ: What is the difference between streaming and progressive downloads?

Streaming

A *streaming* download is a "play-as-you-download" format. It is best suited for business presentations and online learning. Streaming is the only way to deliver live feeds and support broadcasts and *multicasts*. A *user*'s machine plays data as it is received and then discards it. Whether providing live broadcasts or video-on-demand (VOD) stored on a *server*, streaming allows broadcasts to run as long as needed.

One of the primary goals of streaming video is to maintain smooth real-time playback at various connection speeds. To make this possible, streaming media relies on different protocols and servers for delivery than standard Web pages. Real-time Protocol (RTP) and Real-time Streaming Protocol (RTSP) are known as connectionless protocols, in which speed is a higher priority than accuracy. Streaming servers reduce *bandwidth* overhead by broadcasting data across a network without verifying whether it is actually received.

Streaming software also allows you to adjust the data rate of your broadcast downward to meet the audience's capabilities. Audiences with lower bandwidth connections can still be reached but with some quality compromised.

Note In some cases, streaming servers may have ineffective connection speed detection .The bitrate selection may not always be optimal, the result being that users will sometimes receive streams at a lower quality than necessary, or will receive streams at a higher quality causing stuttered video play.

For a table comparing progressive downloading with streaming, see the FAQ: What is the difference between streaming and progressive downloads?

FAQ

1. **What is the difference between the old Eyeblaster video components and the new unified one?**

   The unified video component unifies the functionality of the FLVProgressiveLoader and FLVStreamingPlayer, video components into one new Video Loader and Video Playback component. Whereas before you had to select the appropriate video component and drag it to the stage of your Flash document, the new unified component lets you select the required video delivery features and define all the user experience settings within the video loader.

2. **When should I use the separate video components as apposed to the unified one?**

   The unified component can be used in most cases except if you want to place the buttons somewhere else other than at the bottom of the video, or if you want to exclude the buttons from the video altogether.

3. **What is a Video Package?**

   A Video Package is a collection of different versions of the same movie in different formats designed to cover the various bitrate requirements and formats of the user. This package is created within the Video Studio, using the required track as part of the process of video creation for optimal user experience. A package is associated with an ad so that all videos are available for all users.

4. **How long does it take to create a package?**

   A package created in 3 basic steps takes about 10 minutes.

5. **What is the difference between the 4 video package creation tracks/delivery options?**

   Each track provides the best packaging option depending on the type of raw video files you have and how you wish to upload them to the system.

6. **How long does it take for Eyeblaster to convert video to FLV and WMV?**

   2-3 business days if the video is digitized. 3-5 business days if the video requires digitization.

7. **What's the difference in the user experience between WMV and FLV?**

   Essentially there is no difference. WMV tends to have better quality and bandwidth detection. If there is a WMV file available, we will try to serve that first. The buttons that are available to the user in full screen mode differ because the WMV plays as a true full page WMV video, whereas the FLV plays in a full page browser

8. **Which Eyeblaster formats can use the Video Studio?**

   All: Floating Ad, Polite Banner, Expandable Banner, Pushdown Banner, Yahoo Floating Ad, MSN Floating Ad, Window Ad, Commercial Break, In Game Ad.

9. **Which video formats are supported?**

   FLV 7, FLV 8 and WMV 9 are all supported.

10. **What do I do if the video is not in a supported format?**

    Take advantage of Eyeblaster offers a free-of-charge service to convert unsupported video formats to FLV and WMV.

11. **Can there be more than one video per ad?**

    Yes, an ad can have up to six packages. Each package should have a different video. An example of using multiple packages would be a movie studio that has three trailers and wants one ad where the user can choose to see any of the three trailers. Each trailer would be one package and one package would have all the videos for the different bandwidths and formats in it.

12. **Is full screen video available?**

    Yes, in both WMV and FLV formats. It is activated on user click.

13. **Do I have to make separate video files for full screen?**

    No, the full screen version of the videos is created automatically by the Video Studio.

14. **At which bitrates does Eyeblaster set high, medium and low bandwidths?**

Eyeblaster has four levels: best is 750K and higher, high is 500K, medium is 300K and low is 100K. These bitrates can be customized in the Video Studio.

15. **Do separate ads need to be made for the different bandwidths?**

    No, the files for the different bandwidths form a package and the complete package is part of the ad, so Eyeblaster will show the ad with the correct video for each bandwidth automatically.

16. **Does the creative shop need to make files for each bandwidth?**

    No, Eyeblaster Video Studio automatically takes care of this, but if the creative shop wants to, they can create different videos.

17. **What happens to users with dial-up or low bandwidth connections?**

    Users with bandwidth under 100K are not able to view full screen video. Users with 56K will not be served video at all.

18. **Does video work with other Eyeblaster features?**

    Yes, dynamic data, data capture, user polling, post-click tracking, synchronized ads and behavioral ads all work. Additionally, all rotation options (weighted, time-based, automatic optimization) are all available. Note, however, that if you want any of these features to overlay the video, you cannot use WMV.

19. **Can there be interactivity in the video?**

    Yes, in an FLV, there can be an interactive layer over the video.

20. **What is automatic bandwidth detection?**

    Automatic bandwidth detection is part of the test performed by video player plugins to determine the user's bandwidth in order to serve the movie at the most suitable bitrate. This is achieved by measuring the time it takes to download the Flash asset that precedes the video. If the video is a WMV, Eyeblaster uses the built-in WMV bandwidth detection feature.

21. **What if some publishers require streaming while others accept progressive?**

    Only one video asset needs to be made. Eyeblaster Video Studio will automatically create a streaming and a progressive version and serve the correct version to the specific publisher.

22. **What tracking and reporting capabilities are available?**

    For both video and full screen video, Eyeblaster tracks the percentage played at 25, 50, 75 and 100%. Additionally, reporting includes the number of users that paused, muted and closed the video.

23. **How is pricing structured?**

    There will be one price for files under 2.2MB and another for files between 2.2-4.4MB. (The video plus the ad is limited to 4.4 MB). Additionally, there will be a price if the video is set to automatically play. Note that pricing is the same for both full screen and regular view.