Bitrix Site Manager 8.x

Video Publishing Guide

Introduction

Video materials offer a great opportunity to broaden the way of information provision. The two special components exist: **Media Player** and **Video Library** to allow users to upload videos for other visitors to play.

This manual is for site developers with the knowledge on HTML, CSS, PHP, JavaScript.

The document describes the main features and modus operandi of the **Media Player** and **Video library** components.

Chapter 1. The Media Player

The **Media Player** component plays video files in the following formats: FLV and WMV. Besides, the component is also capable of playing MP3 and AAC audio files. Another possible application is an image viewer supporting the most of the common image file formats.

The player can play single files or a sequence of files by using playlists or the **Video Library** component.

The **Media Player** component supports the following playlist formats:

- XSPF;
- ATOM;
- RSS;
- ASX.

> **Note. Media Player** requires **Adobe Flash** or **Microsoft Silverlight** to be installed on a client's computer. If none of them is installed, a corresponding notification message will be shown.

The **Media Player** design can be altered using special skins that a user can create personally if required.

Preparing and Uploading Video Files

There exist many video formats; the new ones appear now and then so it is rather hard to support all of them in a little component. That is why you have to convert your video to **WMV** or **FLV** format. You can use any video converter software you like; the only requirement is that it must give you a well-formed converted file. For example, you can try **Afree All Video Converter** (http://www.afreesoft.com).

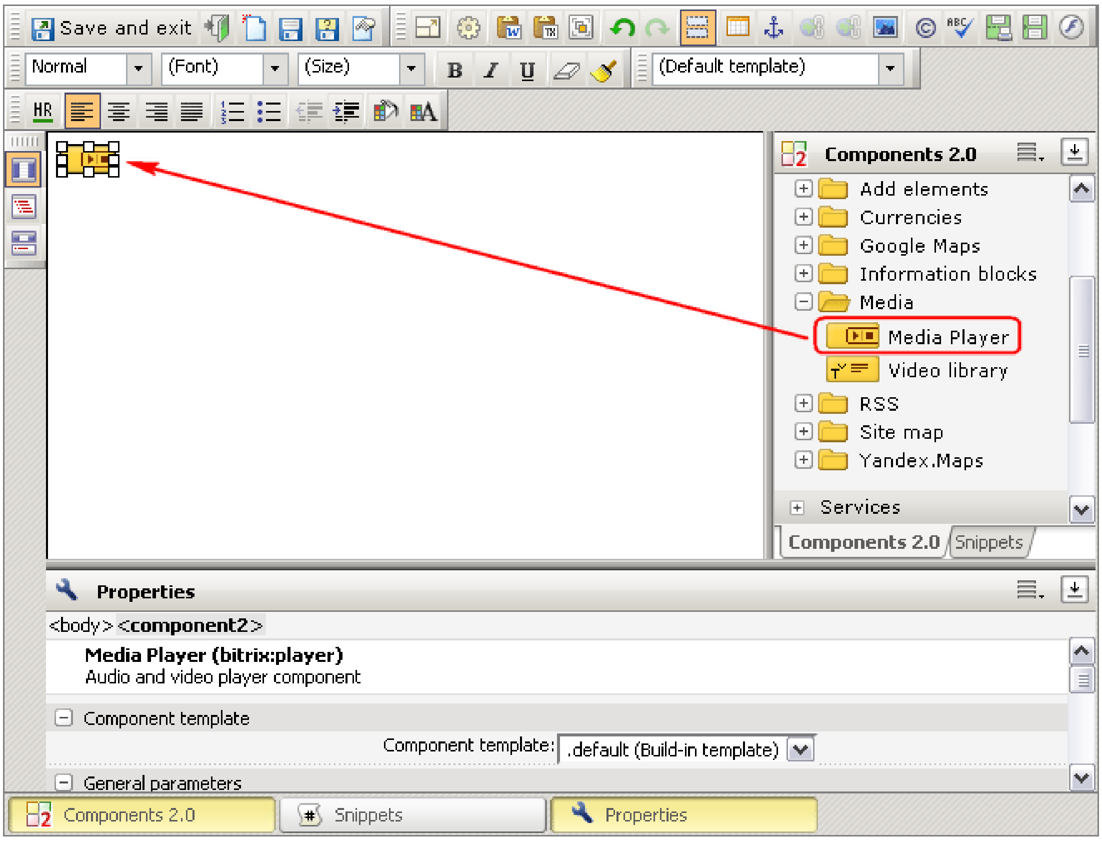
Sometimes, Internet Explorer users may face some problems playing video files not related to the component itself. Internet Explorer imposes queer restriction on the preset cache size which makes viewing large files impossible. It is rather difficult to figure out the optimal file size since the cache size value varies depending on the browser version.

In order to avoid complaints about the video unavailability, split large files into parts. You can split files (as well as optimize them; add subtitles, effects and so on) in any video editor application.

Use **Site Explorer** to create a special folder in which you will save your video files. For example, create the **video_files** folder in the **upload** folder.

Adding and Configuring the Media Player Component

Create a new page and add the **Media Player** component **(bitrix:player)** to it.



Media Player (bitrix:player)
Audio and video player component

Component template

Component template: .default (Build-in template)

General parameters

Fig. 1.1 Adding a component

Note. If a site template has been created manually (without using the visual editor), a situation may occur when the call *<?$APPLICATION->ShowHead();?>* is missing from the *<head>* tag. In this case, the **Media Player** component will not be able to operate. Hence, if you cannot see the component added, make sure this call exists.

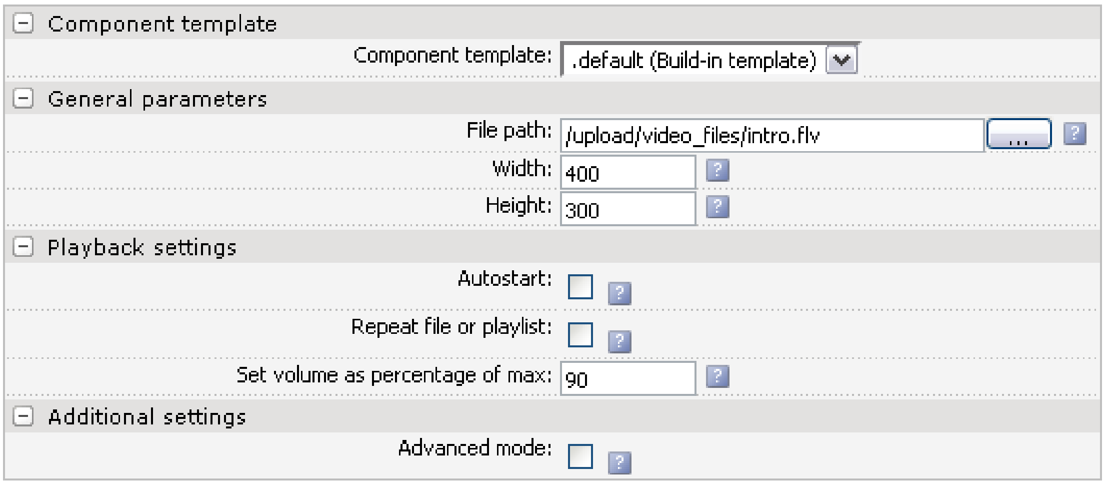
The component can be configured in **simple** or **extended** mode. Simple mode is active by default, while extended mode is recommended for the initial configuration of the component. When you switch back to simple mode, all the extended settings remain.

The modes can be switched between by checking or unchecking the **Enable Advanced Settings** option.

Simple Mode

Note. If the component has been configured to **Use Playlist** in extended mode, the **File Path** field becomes **Play List File Path** in simple mode.

This mode requires that you configure the following parameters:



⊟ Component template

Component template: .default (Build-in template)

⊟ General parameters

File path: /upload/video_files/intro.flv [...]

Width: 400

Height: 300

⊟ Playback settings

Autostart: □

Repeat file or playlist: □

Set volume as percentage of max: 90

⊟ Additional settings

Advanced mode: □

Fig. 1.2 Simple configuration mode

- **File Path** – the path to the media file. Specify it manually or click **[…]**. You can paste the direct download YouTube link in this field.
- **Width** and **Height** are the player window sizes in pixels, respectively.

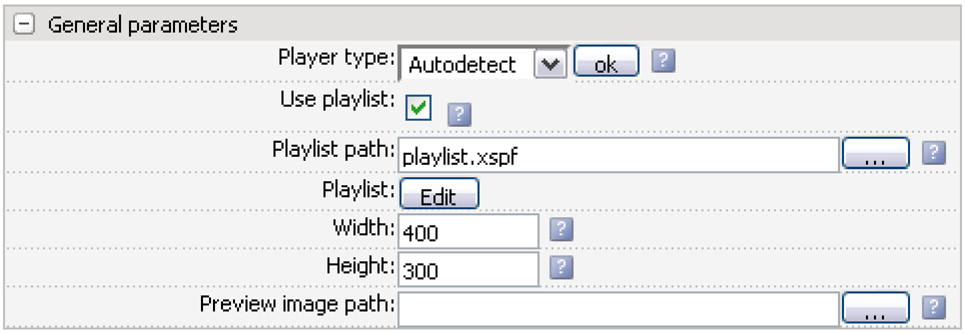
> **Note.** It is recommended to specify the real video dimensions in the **Width** and **Height** fields. Otherwise, the picture will be garbled due to size disproportion. Moreover, add 26 pixels to the **Height** parameter (which is the player control bar height). Another way to avoid garbling is to hide the player controls or select the pop-up panel option in extended mode.

- **Autostart** – if checked, a media file will start playing automatically right after the page is loaded.
- **Repeat file or playlist** – if checked, a media file or a playlist will repeat playing again and again.
- **Set volume as percentage of max** – percentage of a maximum system sound volume.
- **Advanced mode** – switches to extended configuration mode.

Extended Mode

You can switch to extended mode by checking the **Advanced mode** box in simple mode.

The component parameters sheet (fig. 1.3) varies depending on the player type and the play mode selected (single file or playlist).



General parameters

Player type:	Autodetect
Use playlist:	☑
Playlist path:	playlist.xspf
Playlist:	Edit
Width:	400
Height:	300
Preview image path:	

Fig. 1.3 Main settings

- **Player Type** – specifies the player type: **Flash** or **WMV**. If you expect files of different formats, select **Auto Detect**.
- **Use Playlist** – specifies to use a playlist instead of individual files. In this case, provide the playlist path and click **Edit** to arrange the playlist (see the Playlists section).
- **File path** or **Playlist path** – the path to a media or playlist file, depending on the play mode selected.
- **Width** and **Height** – the player window sizes in pixels.
- **Preview image path** – the path to the initial preview image.

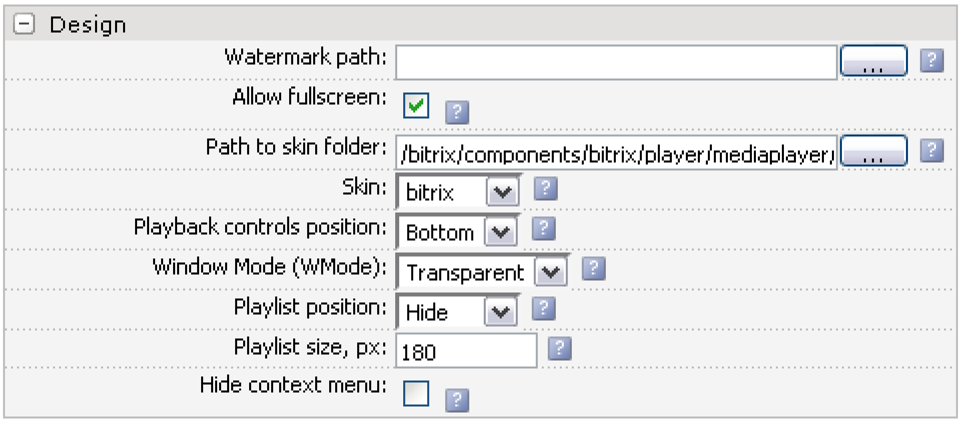
The **Design** group defines the player look and feel and depends on the selected player type and the play mode. The following parameters are common for both players:

Watermark path – the path to an image which will be overlaid on the video. Use this option to add copyrights to your media.

> **Note.** The player does not allow to modify or specify the logo location and the transparency factor. It is best to use a GIF image with transparent background with the same size as the video window. If you specify the logo image, **Preview image path** is optional since the logo will always be visible.

Allow fullscreen – if checked, the player controls will show a button switching to the fullscreen mode.

Here are the **Flash player** parameters (fig. 1.4):



⊟ Design

Watermark path:	
Allow fullscreen:	☑
Path to skin folder:	/bitrix/components/bitrix/player/mediaplayer/
Skin:	bitrix
Playback controls position:	Bottom
Window Mode (WMode):	Transparent
Playlist position:	Hide
Playlist size, px:	180
Hide context menu:	□

Fig. 1.4 The Flash player parameters

Path to skin folder – specifies the skin folder path. The user skins should be stored in a directory that will be never modified by the update system.

Skin – specifies a skin selected from the list. The list is populated with items from the skins folder.

Playback controls position – specifies the player controls location:

- **Bottom** – at the bottom of the window;
- **Over** – the controls toolbar appears at the top of the window when a user hovers the mouse pointer over the video window;
- **Hide** – the toolbar is hidden and a file can be played or stopped by clicking the player window.

Window Mode refers to the Flash player window state:

- **Window** – a media file will play in an individual rectangular window. This allows to render frames at a maximum possible speed;
- **Opaque** – puts the player canvas to the page background;
- **Transparent** – the page background is seen through the transparent areas of a media file. This mode results in slower rendering speed.

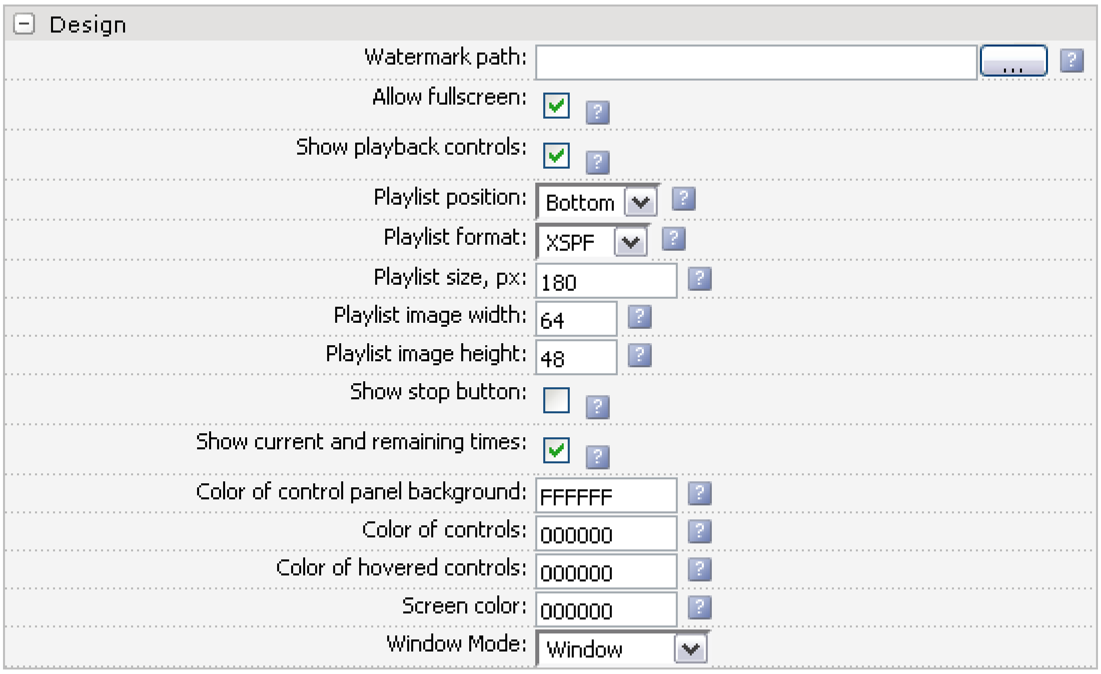
Hide context menu – when checked, the context menu items does not show except for **Settings** and **Adobe Flash Player**.

If the **Use playlist** option is active, you have to configure the following parameters in the Flash player appearance settings:

Playlist position – specify the playlist location, or select **Hide** in order to hide the list.

Playlist size (pixels) – specify the required playlist width (or height, depending on the window disposition).

The following figure shows the **WMV player** parameters (fig. 1.5):



⊟ Design

Watermark path:	
Allow fullscreen:	☑
Show playback controls:	☑
Playlist position:	Bottom
Playlist format:	XSPF
Playlist size, px:	180
Playlist image width:	64
Playlist image height:	48
Show stop button:	□
Show current and remaining times:	☑
Color of control panel background:	FFFFFF
Color of controls:	000000
Color of hovered controls:	000000
Screen color:	000000
Window Mode:	Window

Fig. 1.5 The Windows Media player parameters

Show playback controls – if unchecked, the user will not see the player buttons.

Show stop button – specifies to show the **Stop** button in the player controls.

Show current and remaining times – when checked, the player shows the current and remaining playback time.

> **Note.** The Windows Media player does not support skins. However, you can configure the player colors.

Color of control panel background – specifies the color of the player controls background, as a hexadecimal value.

Color of controls – specifies the color of the player controls, as a hexadecimal value.

Color of hovered controls – the color with which a player control is highlighted when a user passes the mouse pointer over a control.

Screen color – the color of the player screen.

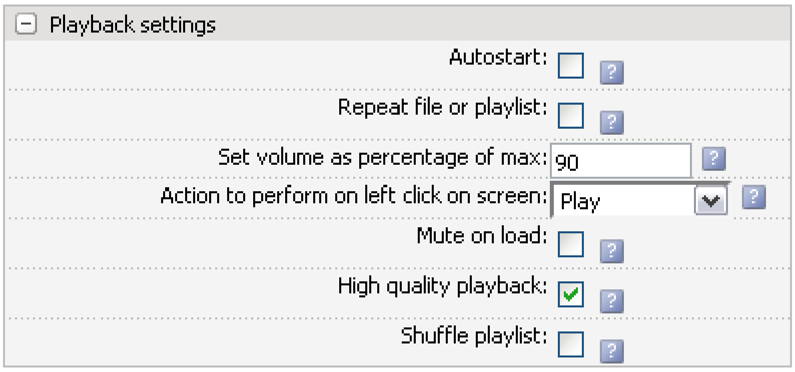
Window Mode – the Windows Media player window mode:

§ **Window** – the player window is hidden behind the page HTML elements;

§ **Transparent** – the player shows inside a page or in a separate browser window.

If **Use playlist** is checked in the component settings, you have to configure the following appearance parameters: **Playlist position** and **Playlist size (pixels)** which specifies the required playlist width (or height, depending on the window disposition). Also, select the **Playlist format** the path to which is specified in the main settings, and the **Playlist image width** and **height**.

The **Playback settings** parameters for Windows Media player are nearly the same as for the simple configuration mode (fig. 1.2). Selecting Flash player adds more fields (fig. 1.6):



⊟ Playback settings

Autostart: □ ?
Repeat file or playlist: □ ?
Set volume as percentage of max: 90 ?
Action to perform on left click on screen: Play ?
Mute on load: □ ?
High quality playback: ☑ ?
Shuffle playlist: □ ?

Fig. 1.6 Flash player settings

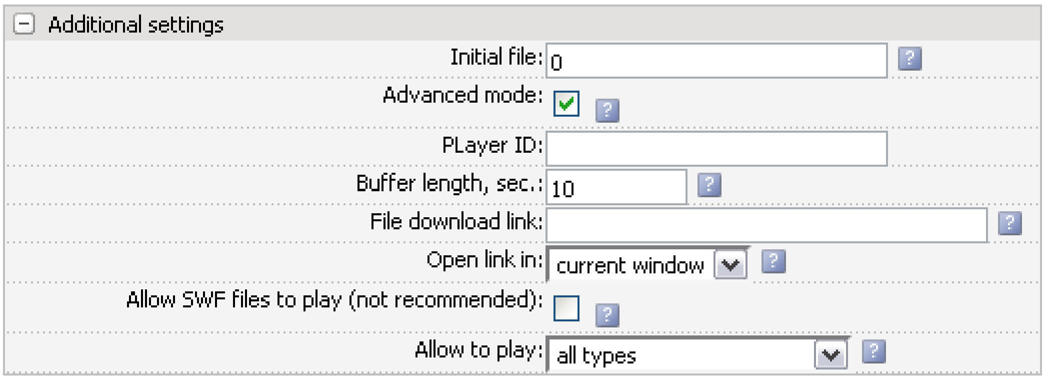
Action to perform on left click on screen – select an action that will be performed when a user clicks on the player window.

Mute on load – if checked, the sound is initially disabled; a user can turn it on manually.

High quality playback –check to play the file at the maximum possible quality. This can slow down the player performance.

Shuffle playlist –plays the playlist entries in random order while the list remains unchanged.

The following parameters exist in the **Additional Settings** group:



⊟ Additional settings

Initial file: 0 ?
Advanced mode: ☑ ?
PLayer ID:
Buffer length, sec.: 10 ?
File download link: ?
Open link in: current window ?
Allow SWF files to play (not recommended): □ ?
Allow to play: all types ?

Fig. 1.7 Additional settings

Advanced mode – switches between the component configuration modes.

Buffer length – specify here how much of a media file the player should download to the buffer before starting the playback, in seconds. The recommended length is 10 to 30 seconds depending on the connection speed. Higher speed needs smaller buffer.

File download link – provide the direct video download link, or the URL to open when a user clicks on the player screen (depends on the **Action to perform on left click on screen** parameter). This option is effective only if playing a single file rather than a playlist.

Open link in – specify a browser window (current or new) in which the download link is to be opened.

If you have configured the Flash player, the following parameters exist:

§ **Allow SWF files to play** – check to allow playing SWF files. Not recommended due to high risk of XSS attack via the SWF file.

§ **Allow to play** – select the content type to play.

If you have configured the Flash player to use a playlist, the following parameter exists: **Initial file**. Specify here the playlist item to be the first to play. Note that the playlist entries are zero based.

Do not forget to save changes you have made to the component parameters. As a result, the public section shows the player according to your settings. For example:

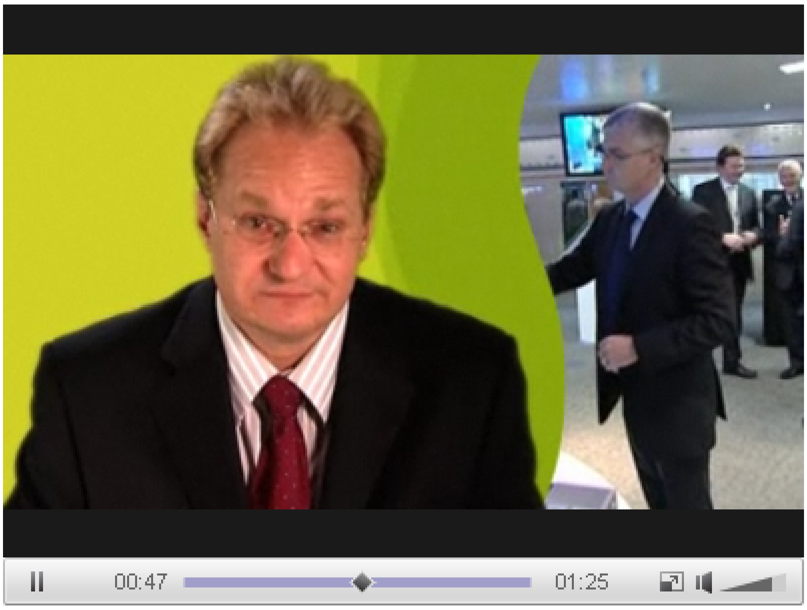


Fig. 1.8 The media player showing in the public section

Using Skins

Only the Flash player supports skins. The **Media Player** component comes installed with the following two skins: **standard** (the default skin) and **bitrix** (developed by Bitrix). However, you can download and use any skin from http://www.longtailvideo.com/addons/skins.

- Open this link.
- Select the required skin and click **Get This**.
- Type in your e-mail. The skin file will start downloading automatically.

As soon as the file has downloaded, you can install the skin.

Note. The archive contains skins for different operating systems. You will need only a **.swf** file. The required file usually has the same name as a downloaded skin, for example: **snel.swf**.

- Move the file to a folder in which you save your skins.

Note. Custom skins can be stored in any subfolder of **/bitrix/components/** or directly in **/bitrix/components/**. The folder **/bitrix/components/bitrix/** is updated by the update system and must not be modified by users.

- Open the **Media Player** component parameters in the extended editing mode.
- In the **Path to skin folder** field, specify the path to the skins folder.
- Select the required skin in the **Skin** folder.
- Save changes.

When the page has reloaded, the player will appear with the new skin applied (fig. 1.9).

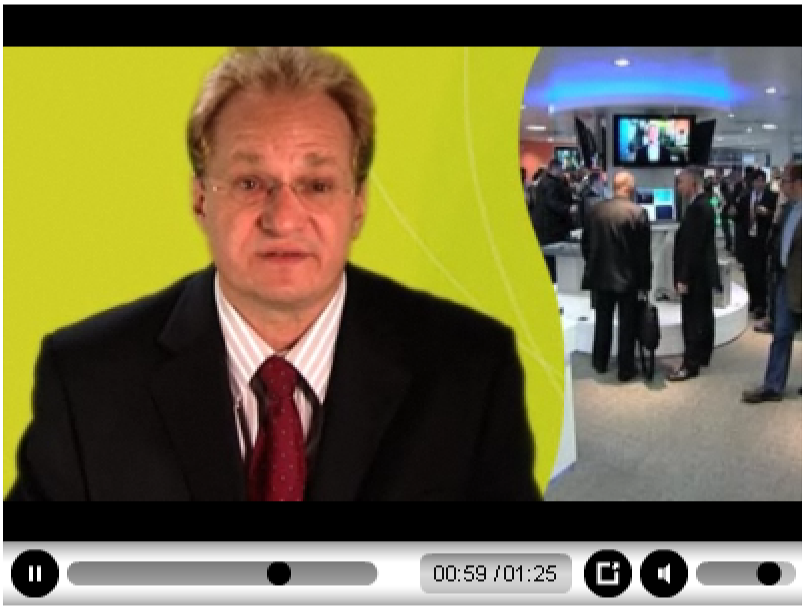


Fig. 1.9 A new player skin

The skins that the producer offers might not fit the design of your site. In this case, you can develop your own skin. For more information, please visit http://www.longtailvideo.com/support/tutorials/Skinning-the-JW-Player.

Adding External Videos

The **Media Player** component allows not just playing video files stored at a local server but also downloading files from the Internet, e.g. YouTube. The following actions are required to be performed on your side to insert an external video.

q Open the component parameters for editing.

q In the **File download link** field, enter any link to an external video. Generally, you can find this link in an **URL** or **Embed** field at a third-party site you want to link to.

q Save changes.

The Playlist

The **Media Player** component can play multiple videos if you have specified a playlist in the player properties. **XSPF** is the most preferable of all supported playlist formats.

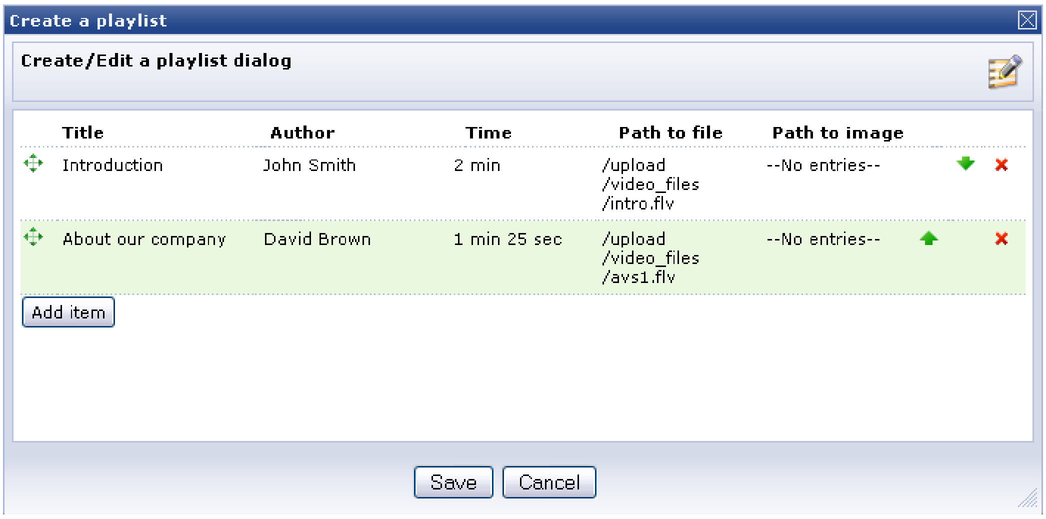
To play multiple videos:

- In the component parameters, check the **Use playlist** option. This will add extra parameters to the **General parameters** group.
- In the **Playlist path** field, specify path to the playlist file manually or using **[…]**.

Note: if a playlist file does not exist, specify the pathname of a new file to be created. Then, create a playlist by clicking **Edit** in the **Playlist** field.

To create or modify the playlist:

- Click **Edit** in the **Playlist** field. The following form will open:



Create a playlist

Create/Edit a playlist dialog

Title	Author	Time	Path to file	Path to image
Introduction	John Smith	2 min	/upload /video_files /intro.flv	--No entries--
About our company	David Brown	1 min 25 sec	/upload /video_files /avs1.flv	--No entries--

Fig. 1.10 The playlist editor

- Add media files. Creating a playlist is generally similar to creating a site menu.
- Save changes.

After the playlist has been created, make changes to the component parameters:

- In the **Playlist Size** field, enter the playlist size adjusted to fit the page design.

Note: the **Height** and **Width** field values include the playlist dimensions. Therefore, add the playlist width or height to set the player size correctly if you use playlists.

- Modify the **Height** and **Width** parameters taking into account the playlist dimensions and location.
- Save changes

Now the player shows the playlist beside:

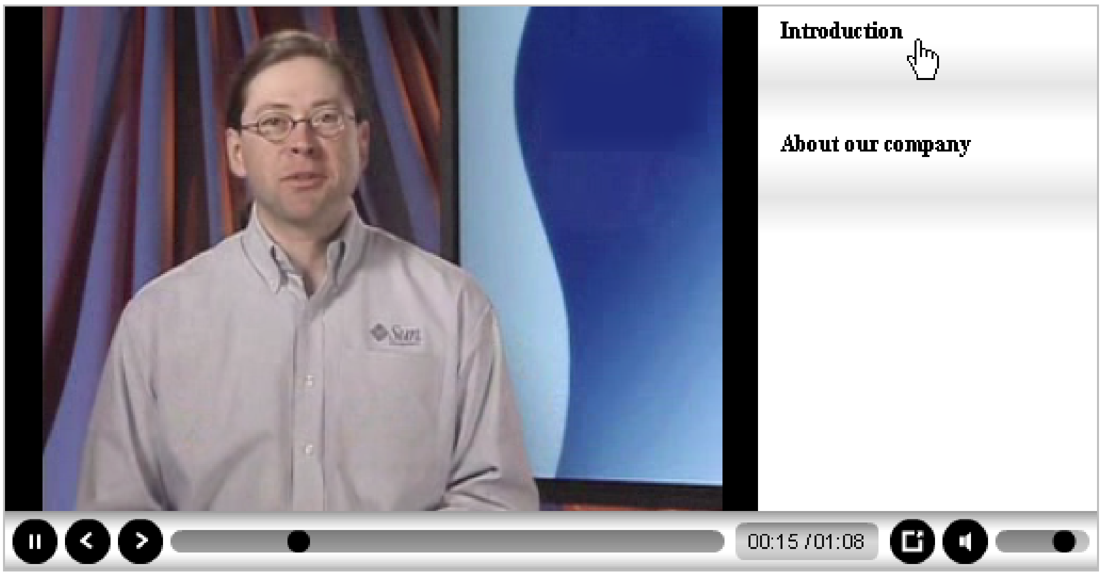


Fig. 1.11 The media player with the playlist attached

Note: the player controls now have the **Next** and **Previous** buttons added.

Not all skins support the display of information existing in a playlist. The above image (fig. 1.11) shows a skin showing the video name only whereas an author and the playtime could also have been added by a playlist creator.

Chapter 2. The Video Library

The video library component functions assisted by the **Information block** module tools. The main purpose for using this component is playing a series of video files. For each video file, an individual information block element exists containing information about the media. The elements must be grouped in sections. The component can play videos of a specific section or of all sections straight away.

Note. For the time being, the component does not support playing videos of a parent information block section.

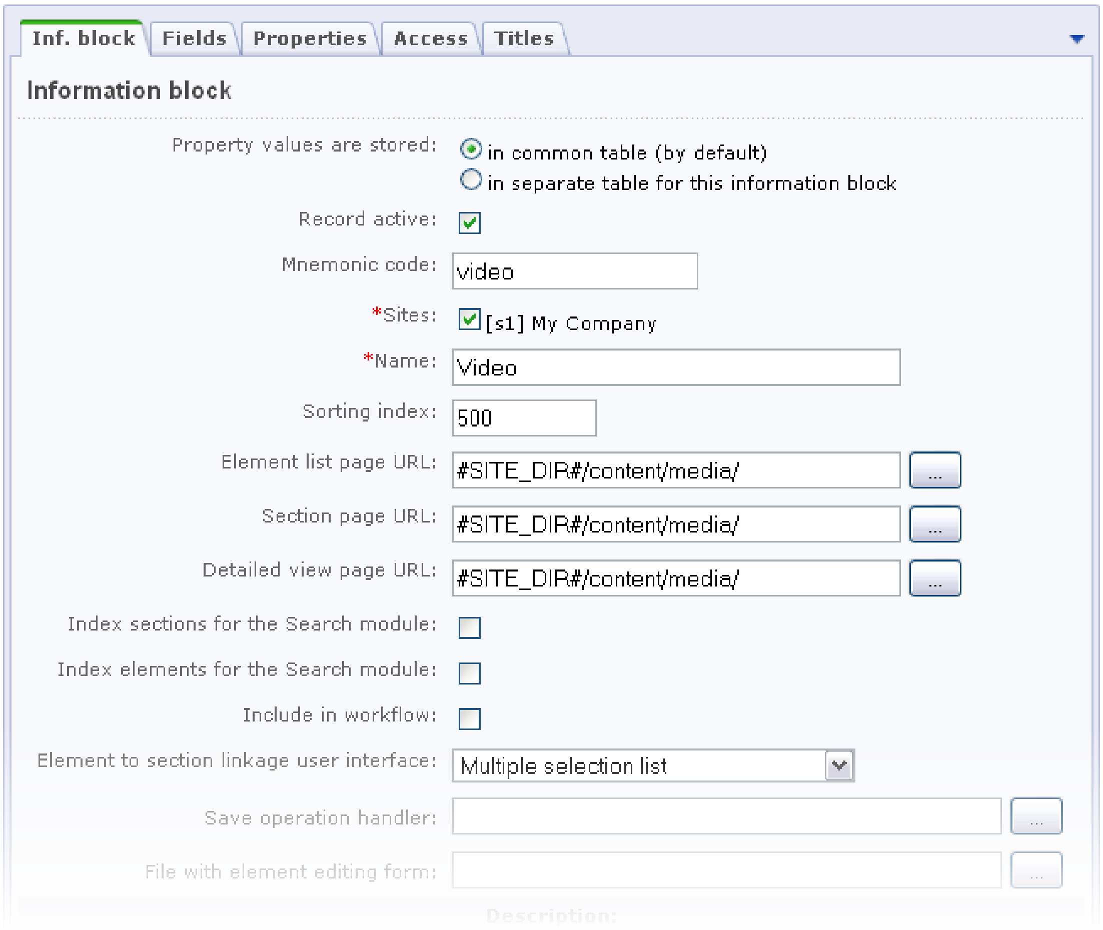
Creating an Information Block

The first step in establishing a video library is to create an information block of any type, say, **Services**.

Note. You can create your own information block type if needed.

Navigate to *Content* > *Inform. blocks* > *Inform. block types* > *Services*.

- Click **Add information block** on the context toolbar. The information block creation form will open (fig. 2.1).



Information block

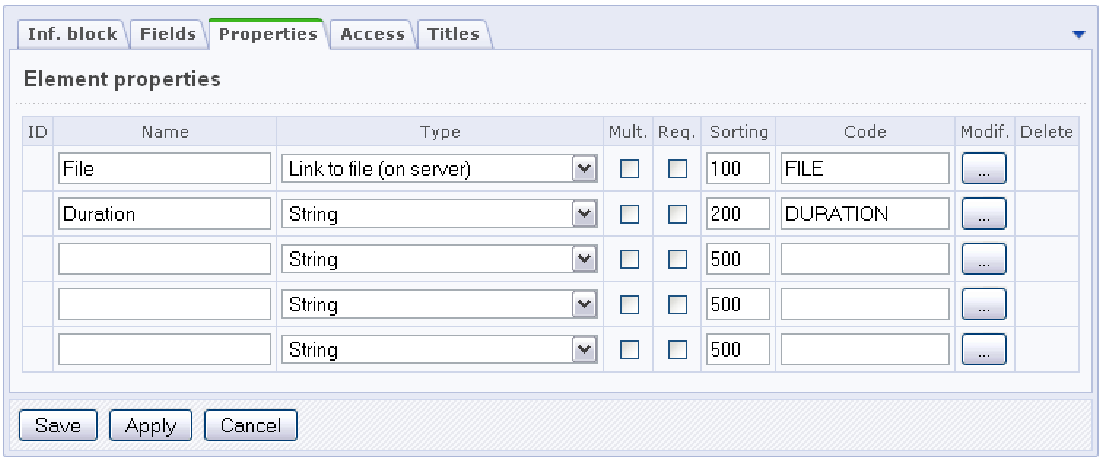
Property values are stored:	● in common table (by default) ○ in separate table for this information block
Record active:	☑
Mnemonic code:	video
*Sites:	☑ [s1] My Company
*Name:	Video
Sorting index:	500
Element list page URL:	#SITE_DIR#/content/media/
Section page URL:	#SITE_DIR#/content/media/
Detailed view page URL:	#SITE_DIR#/content/media/
Index sections for the Search module:	□
Index elements for the Search module:	□
Include in workflow:	□
Element to section linkage user interface:	Multiple selection list
Save operation handler:	
File with element editing form:	

Fig. 2.1 Creating a new information block. The information block editor

q Fill in the **Information block** tab fields:

§ enter the **Mnemonic code**;
§ attach the information block to a required site (or sites);
§ specify the information block **Name**, for example: **Video**;
§ uncheck the search index checkboxes;
§ uncheck the **Include in workflow** option.

q Switch to the **Properties** tab and create the following properties (fig. 2.2):



Element properties

ID	Name	Type	Mult.	Req.	Sorting	Code	Modif.	Delete
	File	Link to file (on server)	□	□	100	FILE	...	
	Duration	String	□	□	200	DURATION	...	
		String	□	□	500		...	
		String	□	□	500		...	
		String	□	□	500		...	

Fig. 2.2 New information block properties

§ a **File** property of a **Link to file (on server)** type. It will contain the path to a media file;

§ a **Duration** property of a **String** type. We will save the video file length here.

q Configure the user group access permissions on the **Access** tab.

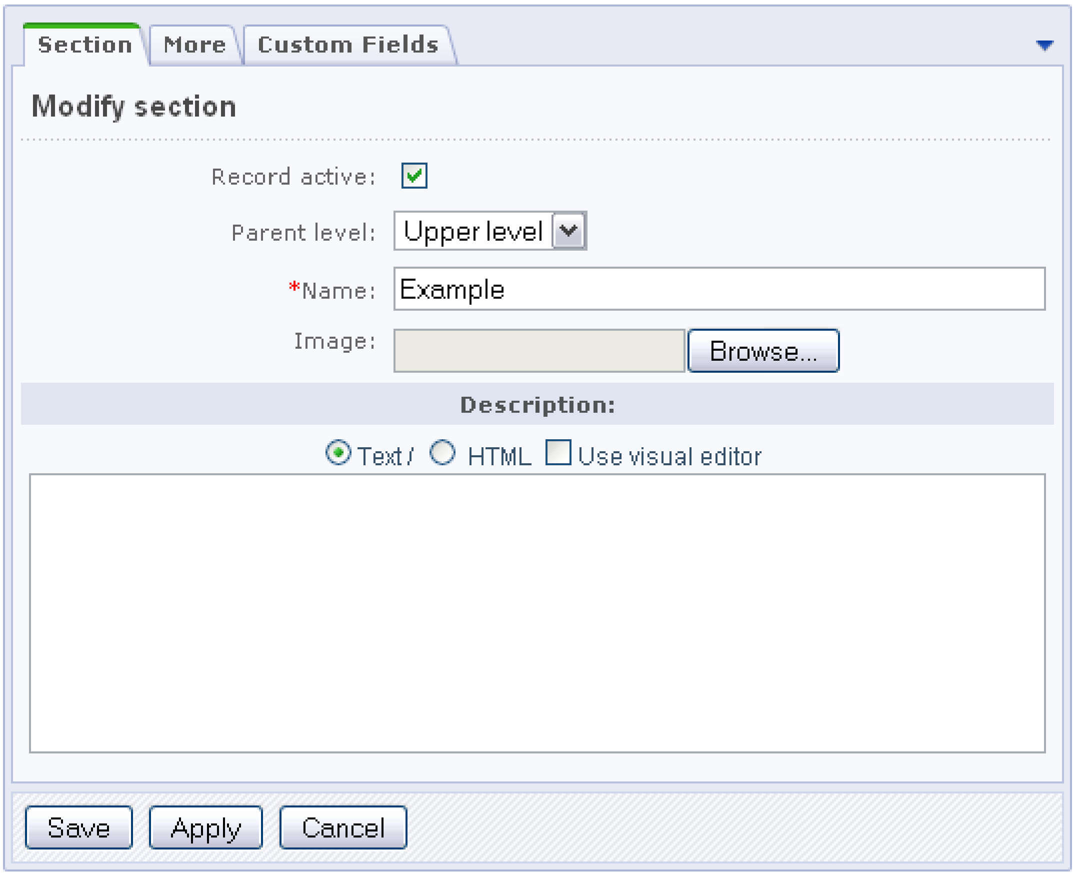
q On the **Titles** tab, edit the captions and titles of the information block entities according to the information block content.

q Save the information block.

Creating Elements and Sections

To create an information block section:

- Navigate to *Content* > *Services* > *Video*.
- Click **Add section** on the context toolbar.



Modify section

Record active: ☑

Parent level: Upper level

*Name: Example

Image:

Description:

● Text / ○ HTML □ Use visual editor

Fig. 2.3 Creating a section

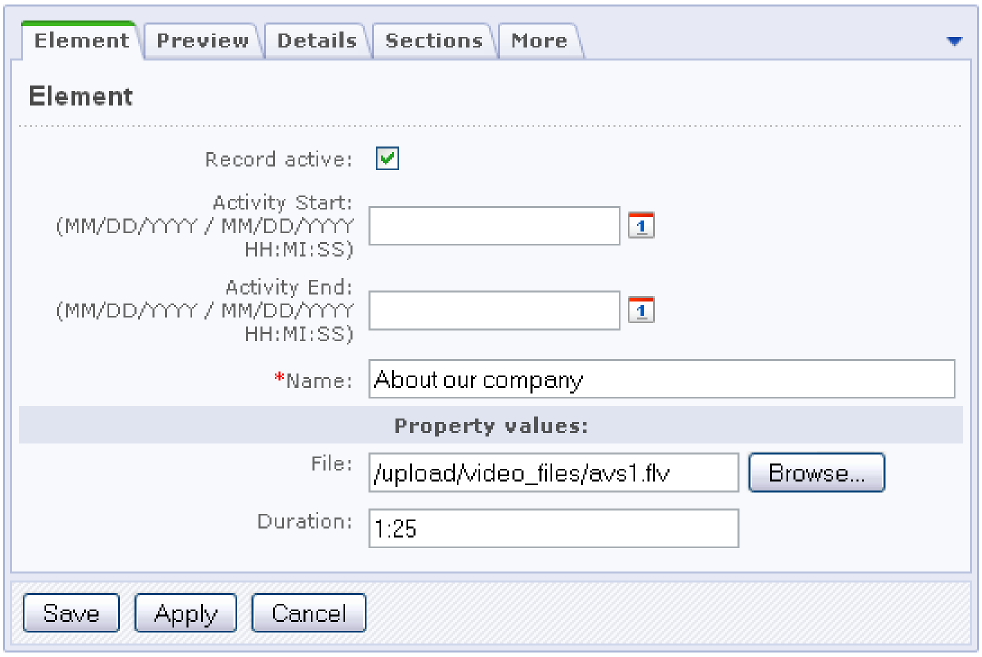
- Enter the section name.
- Click **Save**.

Do the following to add elements.

- Open the required information block section.

Note. Videos stored in the root level of an information block cannot be played.

- Click **Add element** on the context toolbar to open the following form (fig. 2.4):



Element

Record active:	☑
Activity Start: (MM/DD/YYYY / MM/DD/YYYY HH:MI:SS)	
Activity End: (MM/DD/YYYY / MM/DD/YYYY HH:MI:SS)	
*Name:	About our company
Property values:	
File:	/upload/video_files/avs1.flv
Duration:	1:25

Fig. 2.4 Adding a video

To make the video file visible in the playlist, check the **Record active** option. The **Video library** component does not check the element active period; it only shows the element if the latter is active.

- Specify the video **Name**.
- In the **File** field, specify the video file path manually or by clicking **[…]**.
- Specify the video **Duration** if needed.
- Specify **Sorting index** (**More** tab). It acts as a track number to define the position of a video file in the playlist.
- Save the video element.

Possible Issues

You may encounter certain difficulties when creating the information block elements; namely, adding files. The concern is that, while the information block elements are created successfully, the video files are not saved.

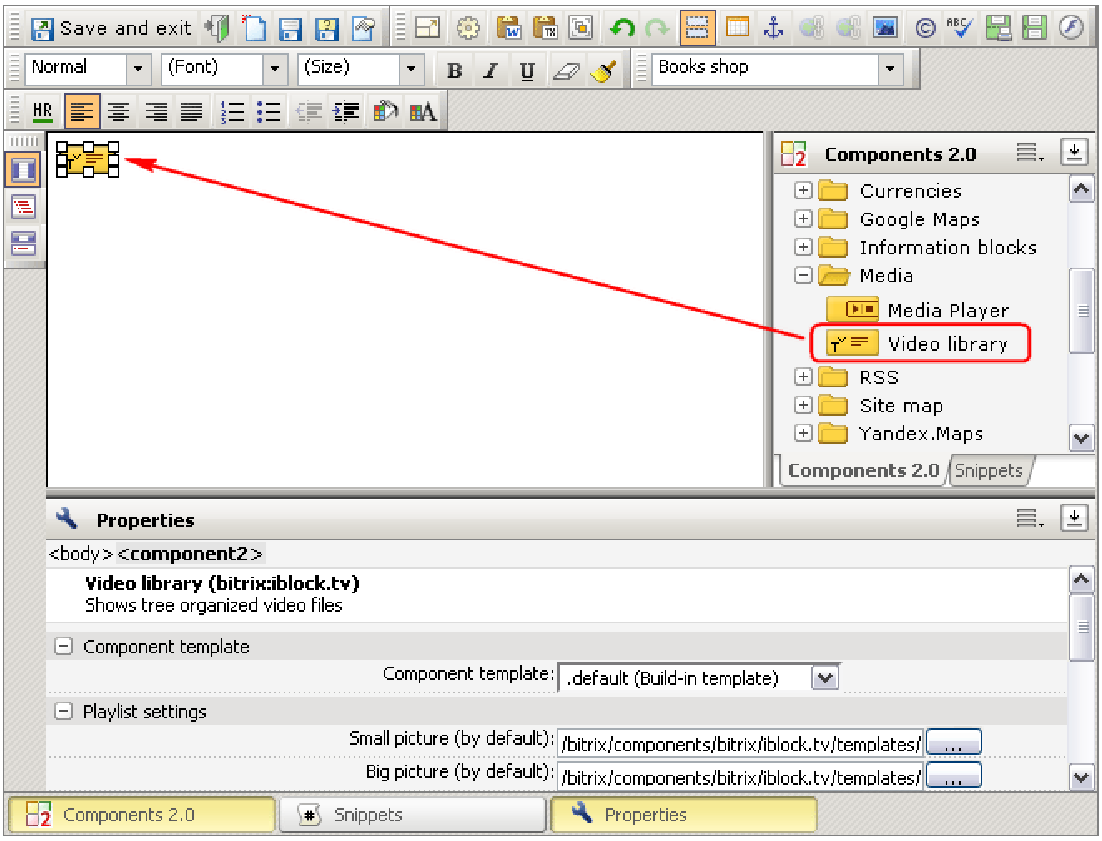
To resolve this problem:

- Type *http://<your_site>/admin/phpinfo.php*. in the browser to open the PHP settings.

- Find the **post_max_size** and **upload_max_filesize** parameters and check their values. They must be no less than the loaded file size (2M is the most common value). If less, ask your hosting service support to increase these parameters.
- If you run Bitrix Site Manager on a local machine, open the **php.ini** file for editing. Increase the **post_max_size** and **upload_max_filesize** parameter values up to, for example, 32 M and save changes.

Adding and Configuring the Video library Component

Create a new page and add the **Video library** component **(bitrix:iblock.tv)** to it:



<body> <component2>

Video library (bitrix:iblock.tv)
Shows tree organized video files

Component template

Component template: .default (Build-in template)

Playlist settings

Small picture (by default): /bitrix/components/bitrix/iblock.tv/templates/

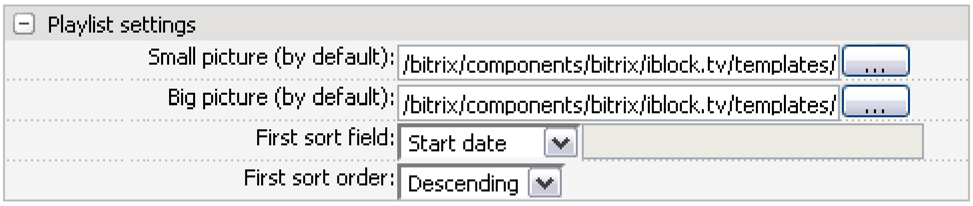
Big picture (by default): /bitrix/components/bitrix/iblock.tv/templates/

Fig. 2.5 Adding the Video library component

> **Note.** If a site template has been created manually (without using the visual editor), a situation may occur when the call *<?$APPLICATION->ShowHead();?>* is missing from the *<head>* tag. In this case, the **Video Library** component will not be able to operate. Hence, if you cannot see the component added, make sure this call exists.

The component includes three templates so that you can configure the component the way you want.

The playlist appearance is configured in the **Playlist settings** parameter group (fig. 2.6):



⊟ Playlist settings

Small picture (by default): /bitrix/components/bitrix/iblock.tv/templates/ [...]

Big picture (by default): /bitrix/components/bitrix/iblock.tv/templates/ [...]

First sort field: Start date

First sort order: Descending

Fig. 2.6 Playlist settings

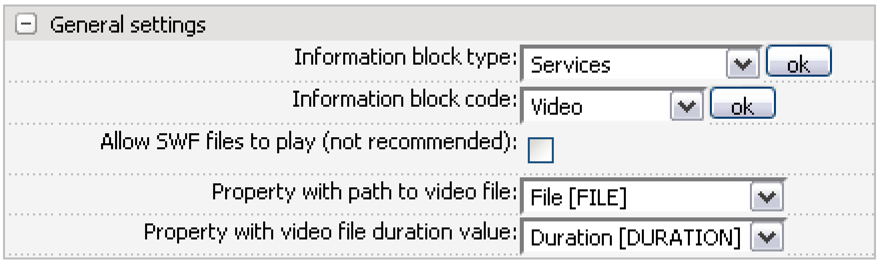
- § **Small picture (by default)** – the path to a preview image. This image will be shown in the playlist.
- § **Big picture (by default)** – the path to a large image associated with the media. The image will be displayed on the player screen before playing the video.

> **Note:** these images are used only when no custom images are specified on the **Preview** and **Details** tabs.

- § **First sort field** – select the parameter by which the videos will be sorted in the playlist.
- § **First sort order** – select the direction in which the files are sorted: increasing or decreasing.

> **Note:** if files with identical names exist in the list, they will be sorted by the system ID's.

As the component functions using the **Information blocks** module, configure the related parameters in the **General settings** group:



⊟ General settings

Information block type: Services [ok]

Information block code: Video [ok]

Allow SWF files to play (not recommended): □

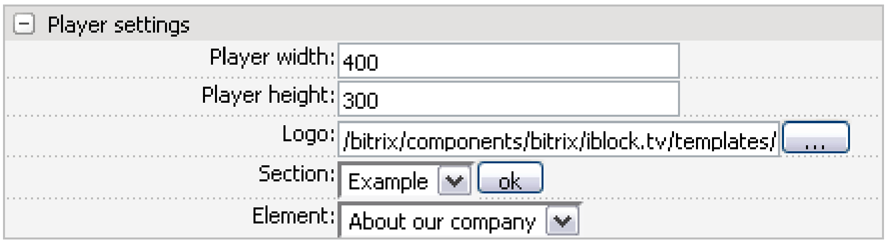
Property with path to video file: File [FILE]

Property with video file duration value: Duration [DURATION]

Fig. 2.7 Information block related settings

- § **Information block type** – the required type of an information block. Select **Services** for our case.
- § **Information block code** – an information block in which the video files are stored; **Video** for our case.
- § **Property with path to video file** – an information block property used to specify the path to the media file.
- § **Property with video file duration value** – an information block property in which the length of a video file is stored.

Leave the default values of the **Cache settings** group parameters and customize the player design in the **Player settings** parameter group:



⊟ Player settings

Player width: 400

Player height: 300

Logo: /bitrix/components/bitrix/iblock.tv/templates/ [...]

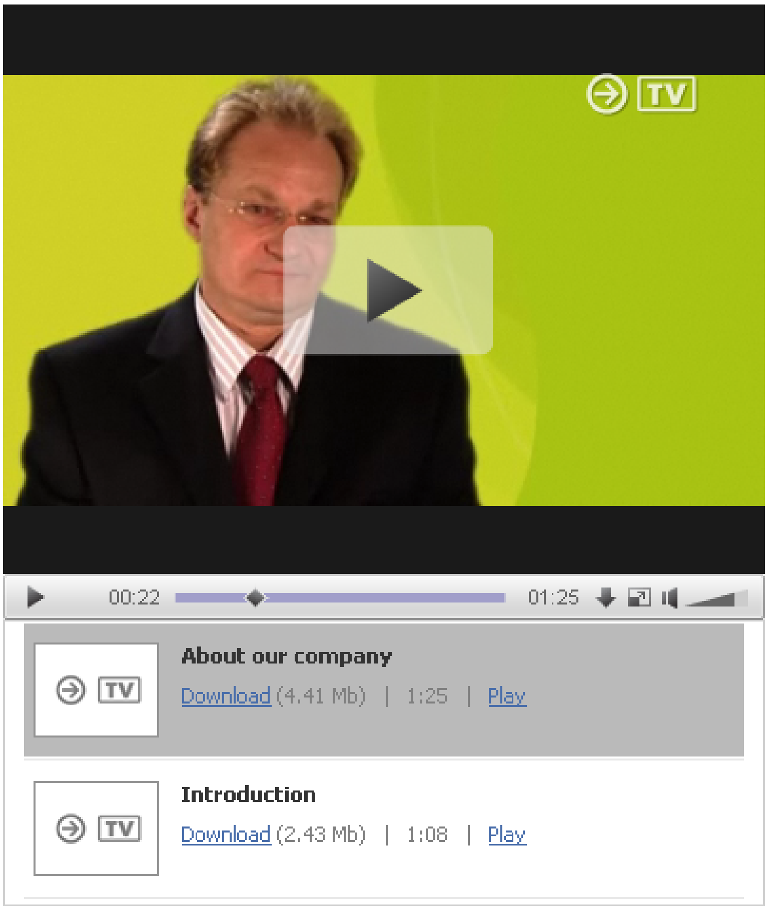
Section: Example [ok]

Element: About our company

Fig. 2.8 The player settings

- **Player width, Player height** – specify the player dimensions.
- **Logo** – the path to an image which will be overlaid on the video. Use this option to add copyrights to your media.
- **Section** – an information block section whose elements are to be played. If selected, the **Element** field becomes available. If you need to play videos from all sections, select an empty value, which hides the **Element** field.
- **Element** – a video file that will be played first by default.

Remember to save your settings. You will see the following result in the public section:



About our company
Download (4.41 Mb) | 1:25 | Play

Introduction
Download (2.43 Mb) | 1:08 | Play

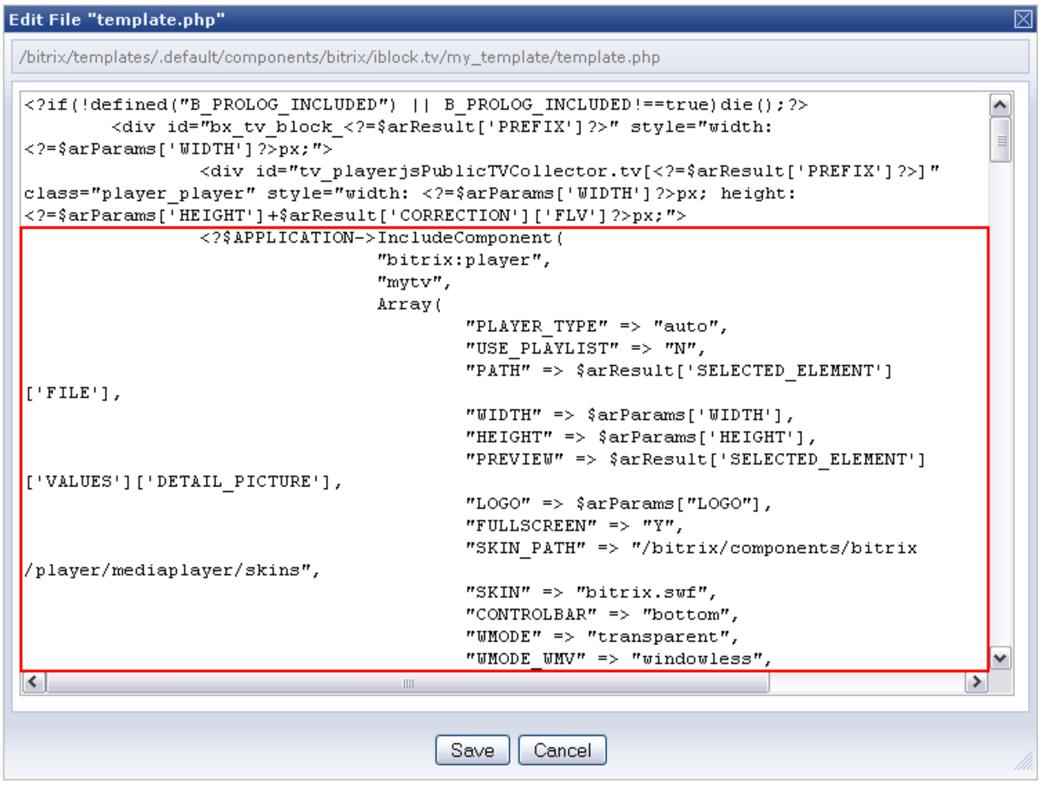
Fig. 2.9 The video library component in action

Additional Settings

The **Video library** component has only a few settings to customize the player design. However, if you want to make more significant changes to the component appearance, do the following:

q Copy the **Video library** component template to one of the site templates, e.g. **my_template** (see also Components 2.0).

q Open the **template.php** file for editing (fig. 2.10).



Edit File "template.php"
/bitrix/templates/.default/components/bitrix/iblock.tv/my_template/template.php

```
<?if(!defined("B_PROLOG_INCLUDED") || B_PROLOG_INCLUDED!==true)die();?>
        <div id="bx_tv_block_<?=$arResult['PREFIX']?>" style="width:
<?=$arParams['WIDTH']?>px;">
                <div id="tv_playerjsPublicTVCollector.tv[<?=$arResult['PREFIX']?>]"
class="player_player" style="width: <?=$arParams['WIDTH']?>px; height:
<?=$arParams['HEIGHT']+$arResult['CORRECTION']['FLV']?>px;">
                <?$APPLICATION->IncludeComponent(
                                "bitrix:player",
                                "mytv",
                                Array(
                                        "PLAYER_TYPE" => "auto",
                                        "USE_PLAYLIST" => "N",
                                        "PATH" => $arResult['SELECTED_ELEMENT']
['FILE'],
                                        "WIDTH" => $arParams['WIDTH'],
                                        "HEIGHT" => $arParams['HEIGHT'],
                                        "PREVIEW" => $arResult['SELECTED_ELEMENT']
['VALUES']['DETAIL_PICTURE'],
                                        "LOGO" => $arParams["LOGO"],
                                        "FULLSCREEN" => "Y",
                                        "SKIN_PATH" => "/bitrix/components/bitrix
/player/mediaplayer/skins",
                                        "SKIN" => "bitrix.swf",
                                        "CONTROLBAR" => "bottom",
                                        "WMODE" => "transparent",
                                        "WMODE_WMV" => "windowless",
```

Fig. 2.10 Editing the component template file

q The template code contains the **Media Player (bitrix:player)** component source code. You will get the desired player look by modifying the player parameters in the code.

q Save changes.

Now, open the **Video library** component parameters for editing. Select **my_template** as the **Video library** component template. Save changes. The player design will change according to your settings.

Final Notes

This manual has given you some insight of using the media related components to publish and play video files on your site.

You can ask your questions at the Bitrix corporate forum:

http://www.bitrixsoft.com/support/forum/,

Should you have any difficulty using Bitrix Site Manager, do not hesitate to send a request to the technical support service:

http://www.bitrixsoft.com/support/index.php