SVP Javascript API for video players

version 1.0
last update 03/02/2011

Concept

The **SVP Javascript API** provide web video publishers a clear and reliable Javascript interface through which their applications are able to control **SVP** video players, get status information and handle player events. Each publisher may implement any part of this **API** interface in his own application/website depending on his needs.

The **SVP Javascript API** is implemented as a set of Javascript methods – Setters, Getters and Listener for handling player events.

The Javascript API interface allow for the following:

1. **Control of video player – play, replay, stop, pause, toggle, adjust, mute, unmute, toggle audio, etc.**
2. **Set video player parameters – volume, videoID, seek to and play/pause, etc.**
3. **Get video player information – is playing, is paused, is complete, is muted, is live, etc.**
4. **Get video player parameters – volume, videoID, stream time/duration, etc.**
5. **Event handling – player started, playhead update/ending/complete, enter/close fullscreen, etc.**

SVP Javascript API is included in standard library of **SVP** players, so it's not necessary to include any additional Javascript file with video player embed code.

NOTE: **SVP Javascript API** is not applicable for EBAY/Yahoo video players.

SVP Player Instance

All Javascript player methods are available through **SVPDynamicPlayer** object. Instance of this object is available from any (except EBAY/Yahoo video players) embed code, associated with some of publisher's videos. Example:

```
...
var vars = {clip_id:"4b3c211a46ef",player_color1:"#c8c8c8",...};
var svp_player = new SVPDynamicPlayer("player4b3c211a46ef", "", "640", "379", {...}, vars);
svp_player.execute();
…
```

NOTE: Javascript player methods are not accessible immediately after page loading, because the actual loading of flash player is shortly thereafter - there is a player event notification (See ***Player Events and Event Handlers*** section).

SVPDynamicPlayer object can be obtained from another global object **SvpPlayerInstances** (also included in standard SVP player library). This object has a method called **getPlayerInstance** which returns **SVPDynamicPlayer** object by **videoID**. Example:

```
var svp_player = SvpPlayerInstances. getPlayerInstance("4b3c211a46ef");
```

It's very useful to make Javascript player method calls in page with more than one videos published.

Player control methods

Example:

```
svp_player.play();
```

method	description
play()	Start video playback.
pause()	Pause video playback.
stop()	Stop video playback.
toggle()	If video playback is started, pause the video. If video playback is paused, start the video.
replay()	Restart video playback from the beginning.
adjust()	Adjust video playback.
toggleAudio()	If audio playback is started, mute the audio. If audio playback is muted, start the audio.

mute()	Toggle volume muting.
unMute()	Increase volume to 100%.

Setters

Example:
svp_player.setVolume(50);

method	description
seekToAndPlay(seconds)	Seek from beginning of the video and start playback. Seek time is specified in seconds.
seekToAndPause(seconds)	Seek from beginning of the video and pause playback. Seek time is specified in seconds.
setVolume(volume)	Set the volume. *volume* has to be an int in the 0-100 range.
loadVideo(videoID)	Load a video whose ID is given by the videoID parameter.

Getters

Example:
var playerVolume = svp_player.getVolume();

method	description
isPlaying()	Return true if player is playing video.
isPaused()	Return true if player is paused.
isComplete()	Return true if video playback is reached end of stream.
isMuted()	Return true if audio playback is muted.
isLive()	Return true if video is live.
getVideoID()	Get the videoID of current playing video.
getStreamTime()	Get the current position in the video in seconds.
getStreamDuration()	Get the video's length in seconds.
getVolume()	Get the current volume setting.

Player Events and Event Handlers

Player events are items that transpire based on a player action. Player events are normally used in combination with functions (event handlers), and the function will not be executed before the event occurs! Player event handlers are registered by player event listener method through **SVPDynamicPlayer** object. For this purpose **addPlayerEventListener** method is provided:

addPlayerEventListener(eventName, eventHandler);

Example:
```
svp_player.addPlayerEventListener('PLAYER.STARTED', function(event){
                alert('Player started successfully!');
        }
);
```

All supported player events are listed in table below:

event name	description
PLAYER.STARTED	This event is triggered when player is loaded.
ENTER_FULLSCREEN	This event is triggered when player enters in fullscreen mode.
CLOSE_FULLSCREEN	This event is triggered when player closes fullscreen mode.
PLAYHEAD.UPDATE	This event is triggered with video playback. *NOTE:* Event can be triggered several times in seconds.
PLAYHEAD.COMPLETE	This event is triggered when video playback is reached end of stream.

PLAYHEAD.ENDING	This event is triggered when video playback is reaching end of stream.
LIVE.CONNECTED	This event is triggered when player is connected to live video stream.
PLAYER_HAS_SLIDES	This event is triggered when player has slides.
SLIDE.CHANGE	This event is triggered when slide has to be changed.