|  |
| --- |
| Dimdim Inc. |
| Console – Publisher Interface |
| Interface exposed by browser controls of Dimdim Publisher |
|  |
| **Bharat Varma Nadimpalli** |
| **1/3/2008** |

|  |
| --- |
| This document outlines the methods exposed by Dimdim Publisher through browser plugins |

getVersion

# Signature

LONG getVersion()

# Description

*Returns the version of the publisher build as a LONG value.*

# Formula

# Usage

1. Publisher version 4.0 built on 31st December 2007 ≈ 40123107
2. Publisher version 4.0 bulit on 3rd January 2008 ≈ 40010308

setProperty

# Signature

LONG setProperty (String propertyString)

# Description

*Parses the propertyString and assigns values to corresponding properties on the publisher. If valid parameters are given in the propertyString, a LONG value of ‘1’ is returned. Otherwise, ‘0’ is returned.*

# Formula

Either single or multiple parameters can be set in a single setProperty call.

*“{<property>:<value>,<property>:<value>,…}”*

Valid properties include the following –

## dcsURL

### Description

The complete URL of the conference server where the console is hosted on, including the action method…

### Required for

Calling the ‘seal presentation’ method on the console after PPTX files are successfully converted and uploaded to DMS.

## dmsURL

### Description

The URL of any valid DMS in the following format –

http://<ip>:<port>/uploadPPTXSlide?meetingID=<>&pptID=<>&pptName=<>

### Required for

This URL is used by the publisher to upload converted PPTX slides to the corresponding DMS, after adding any other required parameters to the URL.

## screenURL

### Description

The complete URL of a valid reflector which is to be used for screencasting, in the following format –

http://<ip>:<port>/dimdimScreenShare/<confkey>/<streamID>/

### Required for

This URL is used by the publisher to publish a screencast from the host machine.

# Usage

1. publisher.setProperty("{screenURL:\"http://jp4.dimdim.com:8088/dimdimScreenShare/b1/s1/\"}");
2. publisher.setProperty("{dmsURL:\"http://ip:port/uploadPPTXSlide?meetingID=<meetingID>&pptID=<id>&pptName=<name>\",dcsURL:\"http://ip:port/dimdim/DMSPresentaion.action?pptName=<name>&pptID=<id>&meetingKey=<meetingID>\",screenURL:\"http://ip:port/dimdimScreenShare/<confkey>/<streamname>/\"}", 0);

getProperty

# Signature

String getProperty (String propertyName)

# Description

*Returns the current value of a specific propertyName, as a String.*

# Formula

A single parameter whose value is required.

"name=\"<propertyName>\""

Valid properties whose values can be retrieved are -

## windowList

This parameter retrieves the current active window handles of the host machine and returns them in a JSON format similar to the following -

{windowList:[{handle:<>,caption:<>},{handle:<>,caption:<>},…]}

## screencastResult

This parameter retrieves the current screencastResult and returns it in a JSON format.

### Description

{screencastResult:”0”} : Initial state, when there is no screencast in progress.

{screencastResult:”1”}: State, when a successful screencast is in progress.

{screencastResult:”2”}: State, when a screencast was closed by user action on the tray icon or by a request from the console.

{screencastResult:”3”}: State, when the publisher is unable to communicate with the reflector.

{screencastResult:”4”}: State, when the window corresponding to the handle that is currently shared is closed.

## conversionResult

This parameter retrieves the current conversionResult and returns it in JSON format. conversionResult returned by the publisher will have a conversionResult:<value> pair followed by data returned by the DMS itself.

### Description

{conversionResult:”0”,…}: Initial state, when there is no conversion in progress.

{conversionResult:”1”, …}: State, when a successful conversion is in progress.

{conversionResult:”2”, …}: State, when a conversion is stopped during progress, either by user action on the tray icon or by a request from the console.

{conversionResult:”3”, …} : Stage, when the publisher is unable to communicate with the DMS, to upload converted slides.

# Usage

1. publisher.getProperty("name=\"conversionResult\"")
2. publisher.getProperty("name=\"screencastResult\"")
3. publisher.getProperty("name=\"windowList\"")

performAction

# Signature

LONG performAction (String actionArgs)

# Description

*Initiates a process to perform the requested action and returns a LONG value of 1 if the arguments are valid.*

*For ‘screencaster’ actions, if the mirror driver is not installed, the return value would be -1. For any other error, the return value will be 0. For successful execution, return value would be 1.*

*For ‘conversion’ actions, successful execution returns 1. For unsuccessful executions, return value is 0.*

*‘mint’ operations refer to Management Interface functionality. Management interface offers starting and exiting the publisher component (tray icon) and the mirror driver installation.*

# Formula

The actions that are currently supported -

1. operation=conversion&action=start&filePath=<filepath>
2. operation=conversion&action=stop&docID=<docID>

*Note that the DMS url set through setProperty includes the docID. However, publisher doesn’t store the docID and for stopping the conversion, docID is required \*only\* for a proper JSON value to be set for conversionResult*

1. operation=screencast&action=share&handle=<handle>

*A handle value of ‘0’ denotes desktop sharing.*

1. operation=screencast&action=stop
2. operation=mint&action=run - Runs the publisher component (tray icon)
3. operation=mint&action=exit – Exits the publisher component, killing any screencaster/conversion processes along with it.
4. operation=mint&action=checkDriver – returns 1 if a valid mirror driver installation is found. Otherwise, returns 0.
5. operation=mint&action=installDriver – invokes the mirror driver installation.

# Usage

1. publisher.performAction("operation=conversion&action=start&filePath=\"D:\\abc.pptx\"")
2. publisher.performAction("operation=conversion&action=stop&docID=i1")
3. publisher.performAction("operation=screencast&action=share&handle=0")
4. publisher.performAction("operation=screencast&action=stop")