Logiciel

Dokuwiki : plugin jcapture

Le plugin **jCapture** permet de faire des captures d'écran ou d'enregistrer l'activité audio et du contenu de l'écran et de les télécharger directement vers le gestionnaire de média DokuWiki sans avoir à passer par un fichier.

Avec jCapture, vous pouvez créer rapidement des pages de documentation pour des logiciels avec une interface utilisateur riche.

- License : LGPL

Introduction

Pré-requis

- Java 6 mise à jour 10 ou supérieure.
- voir la page : https://www.dokuwiki.org/plugin:jcapture
- et http://www.hammurapi.com/dokuwiki/doku.php/products:jcapture:start
- en cas de non fonctionnement, réinstaller java puis jcapture

Installation

Téléchargez et installez le plug-in à l'aide du gestionnaire de plug-in et de l'URL http://www.hammurapi.com/dokuwiki/products/jcapture/jcapture.zip ou http://www.hammurapi.com/products/jcapture/jcapture.zip.

L'applet tente de télécharger dynamiquement certaines ressources. Par conséquent, la base de code de l'applet est définie /somenonexistingcodebase car le serveur Web doit renvoyer le code d'erreur approprié pour que l'applet fonctionne correctement. Si cet attribut n'est pas défini, la base de code par défaut est l'emplacement de la page. Dans ce cas, DokuWiki renvoie «La page n'existe pas» avec le code d'état 200 et l'applet échoue.

Si votre serveur renvoie certaines pages personnalisées au lieu des codes d'erreur appropriés pour les ressources inexistantes, vous devez configurer une URL sur le serveur pour renvoyer les codes d'erreur appropriés et pointer le codebase de l'applet vers cet emplacement.

Building from sources

Below is an Ant build file to compile sources, create a jar file and sign jars. Keystore password is passed to the signjar task through storepass property. This article describes how to create a keystore and a key.

Replace _key_alias_ and _path_to_keystore_ placeholders with real values.

The build file expects sources to be in src folder and dependency jars to be in lib folder.

build.xml

```
    <?xml version="1.0" encoding="UTF-8" standalone="no"?>
    <project default="build">
        <target name="build">
            <delete dir="build" failonerror="false" />
            <mkdir dir="build/jcapture" />
            <delete file="lib/jcapture.jar" failonerror="false" />
            <javac srcdir="src" debug="on"
destdir="build/jcapture">
                <classpath>
                    <fileset dir="lib" includes="*.jar"/>
                    <fileset dir="C:/Program
Files/Java/jdk1.6.0_20/jre/lib" includes="plugin.jar"/>
                </classpath>
            </javac>
            <jar destfile="lib/jcapture.jar">
                <fileset dir="build/jcapture" />
                <fileset dir="src" excludes="**/*.java" />
                <manifest>
                    <attribute name="Vendor" value="Hammurapi
Group" />
                    <attribute name="Bundle-Version" value="0.1"
/>
                    <attribute name="Bundle-SymbolicName"
value="com.hammurapi.jcapture" />
                    <attribute name="Class-Path" value="apache-
mime4j-0.6.jar commons-codec-1.3.jar    commons-logging-1.1.1.jar
httpclient-4.0.1.jar httpcore-4.0.1.jar httpmime-4.0.1.jar " />
                </manifest>
            </jar>
            <delete dir="build/signed" failonerror="false"/>
            <mkdir dir="build/signed"/>
            <signjar
                destDir="build/signed"
                alias="_key_alias_"
                keystore="_path_to_keystore_"
                storepass="${storepass}"
                preservelastmodified="true">
                <path>
```

```
                    <fileset dir="lib" includes="*.jar" />
                </path>
            </signjar>
            <property name="releaseDir" value="release" />
            <delete dir="${releaseDir}" failonerror="false" />
            <mkdir dir="${releaseDir}"/>
            <zip destfile="${releaseDir}\jcapture.zip">
                <zipfileset prefix="jcapture/" dir="${basedir}"
excludes=".classpath, .project, bin/**, build/**, wink, build.xml,
lib/**" />
                <zipfileset prefix="jcapture/lib"
dir="${basedir}/build/signed" />
            </zip>
        </target>
    </project>
```

Configuration

Les options de configuration sont conservées dans un fichier local du répertoire de l'utilisateur et sont chargées au démarrage de l'applet.

Video

Options on this tab apply to the video part of recordings. Some options also apply to screenshot capturing.

Frames Per Second
: frames per second during recording. If this option is set to a too high value, jCapture will record as fast as it is possible and then will use actual FPS, i.e. number of frames divided by recording time, during movie rendering.

Border
: if true then recordings and captures will have a 1 pixel grey border.

Toolbar
: if true then recordings will have a toolbar with a start/stop button, progress bar and volume control.

Image Format
: applies to captures and recordings. Any Java-supported image format for captures, JPEG, PNG, or BMP for recording.

Mouse
: if true, then mouse movements will be recorded and shown in the movie. jCapture records only mouse movements, but it doesn't record clicks/double-clicks and it doesn't record mouse pointer shape and uses a pre-defined mouse pointer shape.

Loop
: if true, the movie does not stop at the end but cycles to the beginning.

Play
: if true, the movie does not stop in the first frame, but starts playing after loading.

Video format

select video format, see below how to add video encoders to jCapture.

Scaling

Graphics
: Scale factor for captured images (capture and recording modes).

Speed
: Scale factor for the movie speed, i.e. how much faster the movie shall be comparing to recording time. E.g. if recording FPS is 10 and scaling is 150%, then movie FPS will b 15. Speed scaling is enabled only audio recording is disabled.

Inactivity processing
: this panel is enabled only when audio recording is disabled.

Remove inactivity
: if true, then inactivity (frames where there are no visual changes and no mouse movements withing recording rectangle) are discarded.

Inactivity interval
: time interval since last activity after which inactive frames shall be discarded.

Audio

This tab applies only to recordings.

Record sound
: if true, recordings will contain sound.

Sample rate
: Sound sample rate.

Stereo.
: Sample size in bits.

Source
: sound source.

WAV2MP3 command
: jCapture records sound in WAV format.
To convert sound to MP3 provide a command line which takes two arguments {0} as input file (wav) and {1} as output file (mp3).
If you don't have a command-line mp3 encoder, take a look at Lame, binary builds are available at RareWares.

Editor

Post-recording editor allows to delete “noise” from the movie such as:

- Periods of inactivity, e.g. during long installation or download steps.
- Frames with intermediate UI state, e.g. web page loading.
- Unrelated frames, e.g. user had to temporarily switch to another app in the middle of recording.

When user clicks on “Stop” button, jCapture offers to edit the recorded movie before uploading. If user clicks “Yes” then a movie editor dialog opens (see below).

The dialog consists of the following parts:

Frame image

Shows frame which has focus in the timeline. Frame image scales when editor is resized, preserving aspect ratio.

Timeline

Timeline has two rows:

Screen row has proportions of the screen and shows mouse position on the screen. If mouse position has changed comparing to the previous frame, it is shown as a black dot. If it didn't change, then it is shown as a gray dot.

Audio row shows normalized audio volume level in logarithmic scale, 2 dB per pixel.

Hovering mouse over the timeline opens a tool-tip with a thumbnail image of the frame under the mouse pointer.

Timeline columns are shown with white background for active frames (mouse moved or screen changed) and gray background for inactive frames. It allows easily spot inactivity periods.

Frames in the timeline can be selected with mouse or keyboard (shift-arrow).

Frames marked for deletion are shown with a little red X in the screen cell and with sound volume rendered in gray instead of blue.

Single frame deletion state can be toggled by double-clicking on the frame column in the timeline.

To delete/undelete a group of frames, select them, right click, and then select “Delete frames” or “Undelete frames” in the context menu.

Volume normalization

If “Normalize volume” check box is selected, then recorded audio volume is normalized to make the highest amplitude in the recording to be 0.95 of maximum possible amplitude for the given audio format.

Number in parentheses after “Normalize volumen” shows volume increase in decibels if normalization is applied.

Splash frame

One of frames in the movie (including frames marked for deletion) can be selected to be the “spash frame”. The splash frame is the first frame in the movie which is shown as a still picture on a web page, before movie starts playing. Splash frame allows to communicate the main point of the movie as a picture.

To select the splash frame, click on a frame column in the timeline, right-click and check “Splash” checkbox menu item. To un-select uncheck the Splash menu item or select another frame as the splash frame.

The splash frame is decorated with a little green rectangle in the timeline.

Playing movie

Movie editor can play a movie from the current position or play current selection. To play from the current position, select a single frame, right click and choose “Play” from the context menu. To play a range of frames, select several frames, right click and select “Play”.

To stop playing click on the screen or on the timeline. When playing finishes, the movie returns to the focus frame. Installation and applet permissions

jCapture applet jars are signed and trusting the applet content is all you have to do to make it work - just click the Run button (see below).

If you run into permission problems (which should never happen), you may need to grant the following permissions to the applet codebase as shown below

```
grant codeBase "http://localhost/dokuwiki/lib/plugins/jcapture/lib/-" {
  permission java.awt.AWTPermission "setWindowAlwaysOnTop";
  permission java.net.NetPermission "getProxySelector";
  permission java.awt.AWTPermission "createRobot";
  permission java.awt.AWTPermission "readDisplayPixels";
  permission java.net.SocketPermission "*:*", "connect, resolve";
};
```

Replace http://localhost/dokuwiki with the URL of your DokuWiki installation. The last permission is needed to connect through proxy. If you know address(es) and port(s) of your proxy server(s), then you can replace “*.*” with server(s) IP adress(es) and port(s).

You can use policytool or edit .java.policy file directly.

Image format

Default image format is PNG. You can change image format either through the Options dialog or by changing file extension of the capture. The plug-in capitalizes the extension and passes it as image format parameter to javax.imageio.ImageIO.write() method. Therefore, all formats supported by this class are supported by the plug-in. On Java 6 update 21 supported formats are: BMP, JPG, JPEG, PNG, WBMP, GIF. We tested gif, jpeg, and png extensions - it works fine!

Utilisation

Une fois le plug-in installé, un nouvel élément de la barre d’outils apparaît, comme indiqué ci-dessous.

Lorsque vous cliquez sur le bouton de la barre d’outils, une fenêtre translucide apparaît [1)].
Sélectionnez «Non» si le navigateur vous demande si vous souhaitez bloquer le contenu non signé.

Déplacez et redimensionnez la fenêtre selon vos besoins: Faites glisser le centre de la fenêtre pour vous déplacer et les bordures pour redimensionner.

Cliquez sur le bouton Options pour configurer les paramètres de capture ou d'enregistrement.

Cliquez ensuite sur Capturer pour créer un instantané ou sur Enregistrer pour enregistrer une vidéo flash de l'activité à l'écran.

Une boîte de dialogue de nom de fichier apparaîtra avec le nom de fichier généré automatiquement. Modifiez le nom si nécessaire et cliquez sur OK.

L'image ou la vidéo sera téléchargée sur le Web et un lien vers l'image sera inséré à la position du curseur dans la zone d'édition.

Pour les grandes images ou les connexions lentes, une boîte de dialogue de progression apparaît après plusieurs secondes et disparaît une fois le téléchargement terminé.

Si le téléchargement réussit, la fenêtre de capture disparaît.

En cas de problème, une boîte de dialogue d'erreur s'affiche et la fenêtre reste visible pour une autre tentative de capture.

Le clip vidéo ci-dessous a été enregistré avec jCapture.

D'autres exemples de vidéos et d'images capturées sont ici.

Commandes d'enregistrement

Pause
: met en pause l'enregistrement.

Stop
: arrête l'enregistrement, convertit les captures d'écran et l'audio enregistrés en fichier SWF, les télécharge sur DokuWiki et insère une balise multimédia au niveau du curseur.

Cancel
: annule l'enregistrement.

Troubleshooting

If the plug-in doesn't work, check Java console and Error console (if you are using FireFox). You can also use a web debugger like Fiddler to trace interaction between the browser/applet and the server.

Java 7 tries to download jcapture.jar.pack.gz before downloading jcapture.jar. It also applies to other

jars. Some hosting providers (e.g. 1and1) return code 300 - Multiple choice in this case, instead of code 404 Not found. In this case JCapture doesn't work (it's Java's “feature”, not JCapture's). One solution to the problem is to add CheckSpelling Off to .htaccess file - it works with 1and1. Another possible solution is to rename/convert .jar files to .jar.pack.gz files. Tips & Tricks

If you use plugins..:..:dokuwikilogiciel:internet:dokuwiki:pluginsfr:logiciel:internet:dokuwiki:plugins

plugins

tag to embed down-scaled videos into the page, adding &direct parameter will create a direct link to the file. This approach works well with Image Box plug-in.

Memory issues

During recording jCapture caches audio on disk, screenshots are stored in memory using soft references and in a temporary file from which screenshot images are loaded if garbage collector clears image reference. Composition of SWF file happens in-memory. Therefore, it is possible to run into OutOfMemoryError that for long recordings. If it happens, go to the Control Panel, open Java configuration and add -Xmx<max heap size> runtime JVM parameter. For example -Xmx800m sets the maximum heap size to 800 Megabytes.

You can check amount of free/max memory available to the applet JVM in jCapture Options/About tab:

Video encoders

jCapture comes with SWF video encoder. Additional encoders can be added in the following way:

- Implement com.hammurapi.jcapture.VideoEncoder
- Register your implementation as a VideoEncoder service provider, see java.util.ServiceLoader.
- Add jar file with your video encoder to jCapture classpath. It can be done by modifying applet.php

It seems that an AVI encoder shall be relatively easy to implement with Monte Media Library. Take a look at the Main class in AVIDemo.jar The encoder shall reconstruct screenshots from delta images by writing deltas on the BufferedImage and then write the image to AVI output as shown in the demo. TO-DO

Modify the php code so the applet classpath can be managed from the DokuWiki control panel. This would allow to add video encoders by adding jar's to the classpath instead of modifying applet.php.

Classpath entries shall be resolvable relative to the jCapture lib folder (relative to the main jar). In this case encoders can be distributed as dokuwiki plugins, e.g. for AVI it'd be jcapture.avi plugin. Such plugins will have just jar files and will be referenced by jCapture classpath, e.g. ../../jcapture.avi/lib/jcapture-avi-encoder.jar. The main jar in the encoder plugin shall have Class-Path entry in its manifest file referencing other dependency jars. Integration

jCapture can be integrated with other web applications, e.g. other Wikis.

There are two approaches:

- Java - write Java code to make the applet work with a particular web application.
- Web-app language (e.g. PHP) - write an adapter/plug-in for the web application to work with jCapture applet.

These two approaches are explained below. Web application

In this case the applet tag can be customized to furnish jCapture applet with upload URL and pass-through parameters which are passed to the upload request. Applet parameters

uploadUrl
: URL to upload file.

sectok, opaque
: these parameters are “pass-through” - their values, if provided, are passed to the upload request.

Additional upload request parameters

ow
: always “1”;

Filename
: file name without namespace, i.e. the part after the last colon.

ns
: namespace, the part before the last colon, if filename contains colon, colon otherwise.

Filedata
: binary file data.

Java

In this case integration is achieved by creating a subclass of AbstractCapturingApplet and implementing HttpUriRequest createRequest(String fileName, InputStreamBody bin) and String bodyName(String fileName) methods.

Below is DokuWiki implementation of createRequest() method for your reference.

```
String uploadUrl = getParameter("uploadUrl");
if (uploadUrl==null || uploadUrl.trim().length()==0) {
    String dokuHost = getParameter("host");

    if (dokuHost.toLowerCase().startsWith(HTTPS_PREFIX)) {
        if (dokuHost.lastIndexOf(":")<HTTPS_PREFIX.length()) { // No port
number
            dokuHost+=":443";
        }
    } else if (dokuHost.endsWith(":80")) {
        dokuHost = dokuHost.substring(0, dokuHost.length()-3);
    }
    System.out.println("DokuHost: "+dokuHost);
    StringBuilder uploadUrlBuilder = new StringBuilder(dokuHost);
    String dokuBase = getDokuBase();
    System.out.println("DokuBase: "+dokuBase);
```

```
    uploadUrlBuilder.append(dokuBase);
    uploadUrlBuilder.append("lib/exe/mediamanager.php");
    uploadUrl = uploadUrlBuilder.toString();
}
System.out.println("Uploading to "+uploadUrl);
HttpPost httppost = new HttpPost(uploadUrl);

if (!httppost.containsHeader("Cookie")) {
    httppost.setHeader("Cookie", getCookies());
}

httppost.setHeader("Pragma", "No-cache");

MultipartEntity reqEntity = new MultipartEntity();
String sectok = getParameter("sectok");
if (sectok!=null && sectok.trim().length()>0) {
    reqEntity.addPart("sectok", new StringBody(sectok));
}
reqEntity.addPart("ow", new StringBody("1"));

String opaque = getParameter("opaque");
if (opaque!=null && opaque.trim().length()>0) {
    reqEntity.addPart("opaque", new StringBody(opaque));
}

reqEntity.addPart("Filename", new StringBody(fileName));

int nsIdx = fileName.lastIndexOf(":");
String namespace;
if (nsIdx==-1) {
    namespace = ":";
} else {
    namespace = ":"+fileName.substring(0, nsIdx);
    fileName = fileName.substring(nsIdx+1);
}

if (namespace!=null) {
    reqEntity.addPart("ns", new StringBody(namespace));
}

reqEntity.addPart("Filedata", bin);

httppost.setEntity(reqEntity);
return httppost;
```

Body name method removes namespace part from file name.

```
@Override
protected String bodyName(String fileName) {
    return fileName.substring(fileName.lastIndexOf(":")+1);
```

```
}
```

Such integration requires no changes in the jCapture Java code. See source code for more details.

Désinstallation

Voir aussi

- **(en)** http://
- **(fr)** http://

Basé sur « Article » par Auteur.

1)
It may take some time for the window to appear for the first time because browser downloads applet's jar files.