# JDownload Manual

**Mike Haller**

# JDownload Manual

Mike Haller
Copyright © 2005 Haller Systemservice

# Table of Contents

# List of Tables

# Chapter 1. Introduction

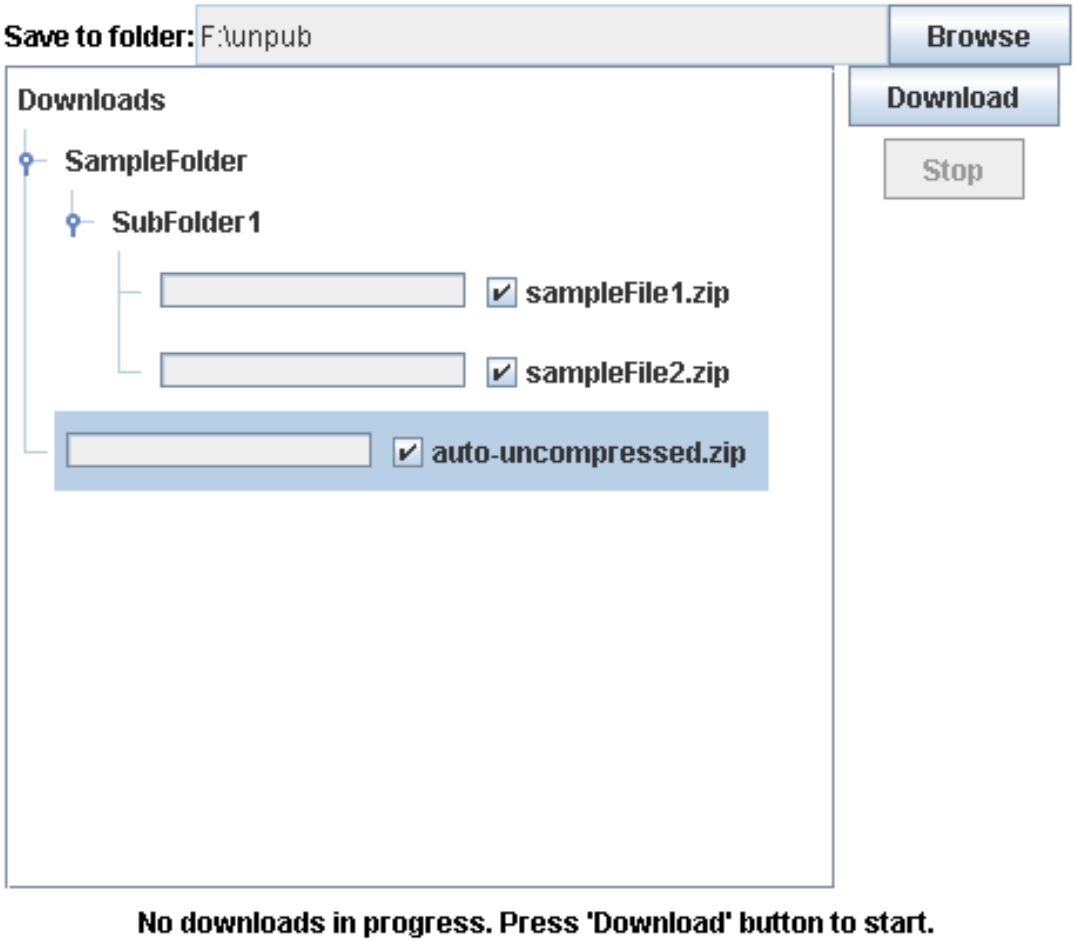
## Table of Contents



# 1. JDownload

JDownload is a tool for multiple downloads from a HTTP server to a user's local file system in one step. It's implemented as a Java Applet and thus platform independent.

The tool is designed for a comfortable, userfriendly interface, so users can easily start using it, without any learning.




Furthermore, the tool can be customized to suit the needs of webmasters: changing colors, the look-and-feel or even use advanced customization. For more details, see the Administrator's Manual ind later chapters.

# 2. System Requirements

Client:

- Requires Java 2 Plug-In, e.g. Sun Java Plug-In 1.4.2 from http://java.com/

  Microsoft Java is *not* supported.
- A compatible Browser (e.g. Microsoft Internet Explorer, Mozilla Firefox)
- Access and write permissions to a target folder
- Internet Connection

Server:

No special requirements on server needed. It must serve the binary JDownload archive (`jdownload.jar`) and (dynamically) create the XML data.

# Chapter 2. Administrator's Manual

## Table of Contents



The Administrator's Manual will give you information about how to install the required files on the webserver and set up a testcase. After editing the XML data file, you will be ready to run a first test with JDownload.

# 1. Tutorial Installation On Server-Side

1. Unpack the release archive into a temporary folder.

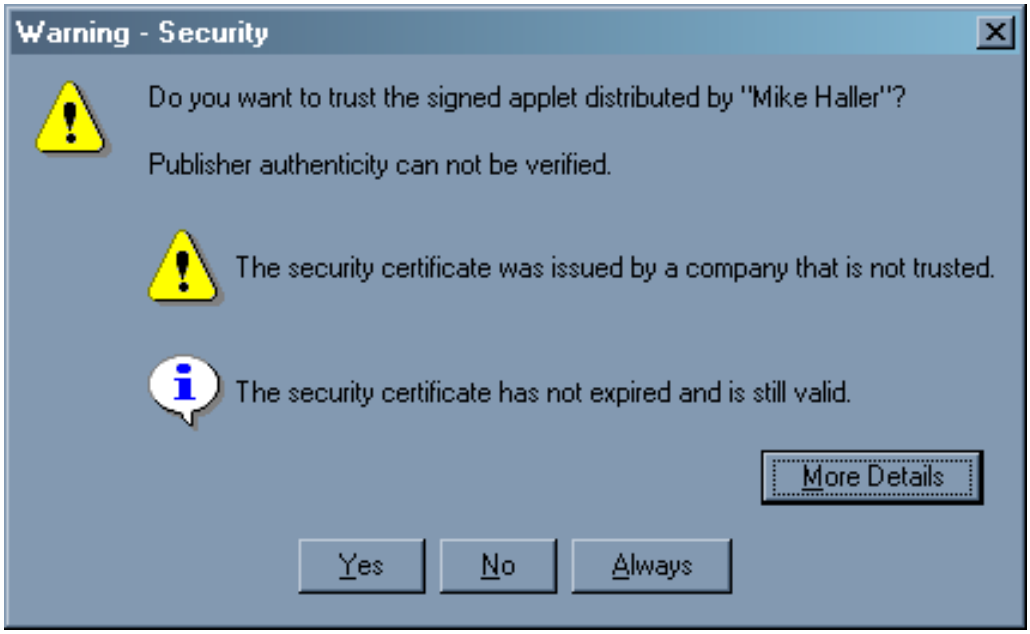
2. Copy the following files into your desired web folder.

   - `jdownload.jar`
   - `download.xml`
   - `index.html`
   - `sample.zip`

3. Start your webserver and open your browser application.

4. Point your browser to the web folder you have copied the files to, e.g. http://www.example.com/jdownload/

   It will then instantly load the JDownload component and show the security dialog.




5. Accept the certificate to start the application

### Important

If you do not accept the certificate, the applet does not have access to your local filesystem, hence cannot save the files or download files from webservers.

# 2. Configuration

Configuring the applet is very easy.

By using PARAM tags, you can setup the preferences:

```
<applet
  code="biz.jupload.jdownload.Manager"
  archive="jdownload.jar"
  width="50%"
  height="50%"
  name="JDownload"
  mayscript="mayscript"
  alt="JDownload by www.jupload.biz">

 <!-- Java Plug-In Options -->
 <param name="progressbar" value="true">
 <param name="boxmessage" value="Loading JDownload Applet ...">

 <!-- An URL pointing to the data structure containing the list of
      files and folders to download -->
 <param name="dataURL"
        value="http://jdownload.jupload.biz/download.xml">

 <!-- Show or Hide the controls
      If hidden (set all to 'false'), remote control the applet
          using JavaScript buttons -->
 <param name="showExplorer" value="true">
 <param name="showControls" value="true">
 <param name="showBrowser" value="true">
 <param name="showStatus" value="true">
```

```
 Your browser does not support applets.
 Or you have disabled applets in your options.
 To use this applet, please update your Java.
 You can get it from <a href="http://www.java.com/">java.com</a>

</applet>
```

| | |
|---|---|
| dataURL | The main setting for JDownload. This value specifies the URL to the XML data file. It can be specified as absolute URL or relative to the codebase. The codebase is the base URL where the applet is loaded froom (e.g. the installation folder)<br><br>It's possible to include query strings. |
| showExplorer | This property shows or hides the tree like Explorer component on the left side of the applet.<br><br>Possible values are `false` to hide the component, or any other value to make it visible. |
| showControls | Use this value to show/hide the buttons Download and Stop on the right side.<br><br>Possible values are `false` to hide the component, or any other value to make it visible. |
| showBrowser | Use this value to show/hide the target folder selection area and the Browse button<br><br>Possible values are `false` to hide the component, or any other value to make it visible. |
| showStatus | Use this to show/hide the status message at the bottom.<br><br>Possible values are `false` to hide the component, or any other value to make it visible. |

The user cannot start the download if you hide the control buttons and do not provide another way (via JavaScript) to remote control the applet.

You can use the JavaScript function `clickDownload()` to start the download process automatically without any user action.

The following setting are dependent on the Java Plugin being used by the user, they are not part of JDownload:

| | |
|---|---|
| progressbar | This is a Java internal setting for the applet loader. It will show a progress bar while the applet is downloaded and started in the browser.<br><br>The applet archive is very small and should not take very long to start. |
| boxmessage | This is a Java internal setting and lets you display a message while the applet is being loaded. |
| image | The image attribute allows you to replace the default coffee cup logo with a custom graphic |

## Note

Due to performance problems with early Java 1.5.0, you should use a different startup `im-`

age, to improve loading performance.

| | |
|---|---|
| boxbgcolor | Sets the background color of the applet box<br>Use HTML-style color values (e.g. #804070) |
| boxfgcolor | Sets the foreground color of the applet box<br>Use HTML-style color values (e.g. #804070) |
| progresscolor | Sets the color of the progress bar<br>Use HTML-style color values (e.g. #804070) |

See this document on how to use the settings with OBJECT and EMBED tags, rather than the APPLET tag: http://java.sun.com/j2se/1.4.2/docs/guide/plugin/developer_guide/using_tags.html

The following attributes of the APPLET tag are support by Java.

| | |
|---|---|
| code | Must be biz.jupload.jdownload.Manager |
| archive | URL pointing to the applet archive, should be jdownload.jar<br>Can have multiple entries, separated by commas, which will be loaded automatically.<br>E.g. you can add the skinlf.jar file here. |
| width | The width of the applet, an absolute pixel value or a percent value, relative to the HTML document width. |
| height | The height of the applet, an absolute pixel value or a percent value, relative to the HTML document width. |
| name | The DOM name of the applet. With this name, you can access the applet from JavaScript. |
| mayscript | Enabled the JavaScript LiveConnect bridge between the Java applet and the JavaScript language.<br>Should be mayscript="mayscript" to enable scripting. |
| alt | Sets a default tooltip for the applet, like with images. |

# 3. Editing XML Data File

At first, you should edit the provided sample file (download.xml) to suit your needs. However, it will run without any modifications. The user will download the provided sample archive (sample.zip) multiple times to different folders and filenames on his local machine.

To edit the XML Data File, you can use a text editor or a XML editor. You can find the specification of the XML structure in the provided DTD file (download.dtd). Use it for validating the XML data file to make sure it can be read correctly by the JDownload tool.

Sample:

```
<?xml version="1.0" encoding="UTF-8"?>
<download>
 <folder name="SampleFolder">
  <folder name="SubFolder1">
   <file name="sampleFile1.zip">
```

```
    <url>sample.zip</url>
   </file>
   <file name="sampleFile2.zip">
    <url>sample.zip</url>
   </file>
  </folder>
  <file name="sampleUncompress.zip" uncompress="true">
   <url>sample.zip</url>
  </file>
 </folder>
</download>
```

As you can see in this example, there is the root element `download`. All other nested elements must be surrounded by the single `download` element.

You can add folders via the `folder` element. The attribute `name` specifies the name of the folder, which will be created on the client side. Folders can have nested subfolders.

Downloadable items are specified by the `file` element, which also has an attribute `name` for specifying the filename on the client side.

Files cannot have any children elements like `file` or `folder`, but they do have a source address. The source URL of the file to download is specified via an `url` element.

The `url` element specifies an absolute URL to the file, but might also be relative to the codebase URL. The codebase URL is the URL where the applet is loaded from.

If, for example, the JDownload component is installed to

`http://www.example.com/downloads/jdownload.jar`

the codebase URL is

`http://www.example.com/downloads/`

# 4. Dynamically Generating XML Data

On server side, you can generate the XML data file dynamically. All you have to configure in JDownload, is to point the dataURL parameter to the URL of the script, which generates the data.

If, for example, you use PHP to generate XML data, you set the dataURL parameter:

```
 <param name="dataURL" value="createxml.php>
```

Within the server-side script, you can dynamically generate the XML structure. For example, you can use a cutomer id or a session identifier to generate personalized download files for the user.

See the sample script `createxml.php` for how to create XML files with PHP.

# 5. Auto-Uncompressing

JDownload is capable of automatically uncompress downloaded ZIP archives on the clients file system.

This can be used to decrease the download time of many files, as compressing them and archiving them into a single file will significantly reduce download times.

To use this feature, add the attribute `uncompress="true"` to the `file` element:

```
<file name="sampleUncompress.zip"
```

```
uncompress="true">
 <url>sample.zip</url>
</file>
```

# 6. Customizing The Component

Webmasters can customize the JDownload applet component to suit their design needs.

The built-in components of JDownload (Explorer-Tree, Control Buttons, Target Folder Chooser) can be disabled and hidden.

If the applet is hidden completely, you need a way to remote control it, so the user can browse for another target folder, start the download progress or stop the download progress.

You can provide this functionality by implementing it in JavaScript. Executing several actions is done by calling the JavaScript JDownload API:

```
<applet
  code="biz.jupload.jdownload.Manager"
  archive="jdownload.jar"
  width="50%"
  height="50%"
  name="JDownload"
  mayscript="mayscript">
```

### Note

The bold tags are important for the JavaScript LiveConnect feature to work. However, you can change the name of the element from **JDownload** to whatever else you want.

This is the corresponding JavaScript code:

```
function openBrowseDialog()
{
 JDownload.clickBrowse();
}
```

For example, it can be called by a button:

```
<input type="button" value="Browse"
            onClick="openBrowseDialog();">
```

- Browse Target Folder

  By calling `clickBrowser()` on the JDownload component, a standard dialog will open and lets the user choose another folder to save the files to.

  This call is a blocking call, which means that the JavaScript call will block until the user returns from the folder chooser dialog with OK or Cancel

- Start Download

  By calling `clickDownload()` the download progress will start. All queued files (files with checks in the checkboxes) will be downloaded to the selected target folder.

  This call is non-blocking, which means the JavaScript call will return instantly to the program flow. The download progress will run in the background, until the user stops the download or the download finishes.

- Stop Download

  By calling `clickStop()` the download progress will halt and completely stop. However, the file which was currently in download progress is saved to the point where it stopped. This makes later resuming possible. After restarting the download (which must not be neccessarily in the same session), the download will restart in the beginning, but will skip downloading previously downloaded file fragments.

  This call is non-blocking and immediately returns. The download will continue for a little time until the next small read block from the network has finished.

# 7. Reacting On Events

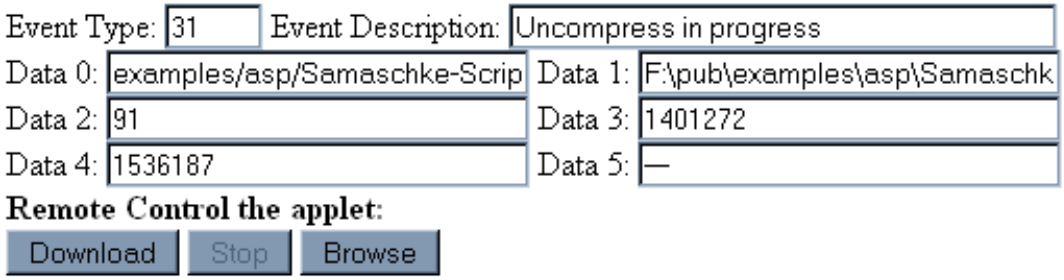
The JDownload component will send event notifications to JavaScript embedded into the same HTML document.

These notifications can be heard by implementing a function called `onJDownloadEvent(EVENT_TYPE, DATA)`

Details about the even types and the data can be found in the Appendix chapter.

The events are fired and sent to the JavaScript function on several actions within JDownload, for example when the user clicks Download, or stops the download.

Status updates can be implemented by retrieving event notification messages of the 'progress' type. These events will hold information about the ongoing process, such as percentage, starting time, ending time etc.




Data 0 shows the file being extracted, Data 2 shows the percentage value (91%), Data 3 shows the current (uncompressed) position in the archive (1.4MB) and Data 4 shows the total length of the uncompressed files (1.5MB).

The buttons can be used to remote control the applet. The HTML buttons are disabled within the JavaScript code, to prevent the user from clicking on buttons, which do not make sense. (You can't stop a download progress, if no download progress is running)

See Section 3, “JavaScript API Event Types” for details of the event types and their argument data.

This is a sample implementation of the event listener function:

```
function onJDownloadEvent(EVENT_TYPE, DATA)
{
 // Display the event type as number and descriptive text
JDownloadMessages.EventType.value=EVENT_TYPE;
  var msg = "---";
  var code = parseInt(EVENT_TYPE);
  switch(code)
  {
   case 0: msg = "Detaching listener"; break;
   case 1: msg = "Attaching listener"; break;
   case 2: msg = "Applet initialized"; break;
   case 3: msg = "Applet waiting"; break;
```

```
   case 4: msg = "Queue status changed"; break;
   case 5: msg = "Target path changed"; break;
   case 10: msg = "Download starting"; break;
   case 11: msg = "Download in progress"; break;
   case 12: msg = "Download finished";
     JDownloadMessages.btnS.disabled='disabled';
     JDownloadMessages.btnD.disabled='';
    break;
   case 13: msg = "Download stopped";
     JDownloadMessages.btnS.disabled='disabled';
     JDownloadMessages.btnD.disabled='';
    break;
   case 20: msg = "Download item starting"; break;
   case 21: msg = "Download item in progress"; break;
   case 22: msg = "Download item finished"; break;
   case 23: msg = "Download item stopped"; break;
   case 30: msg = "Uncompress starting"; break;
   case 31: msg = "Uncompress in progress"; break;
   case 32: msg = "Uncompress finished"; break;
   case 33: msg = "Uncompress stopped"; break;
   default: msg = "Unknown event ["+code+"]"; break;
  }
  JDownloadMessages.EventDesc.value=msg;
}
```

# 8. Look And Feel

The Look'n'Feel of the components can be changed by using the SkinLF package. To configure this, add the skinlf.jar file to the `archive` tag of the applet. The theme must be named `themepack.zip` and reside in the same location as the `jdownload.jar` file.

JDownload will automatically try to initialize the look and feel classes by SkinLF on startup. If this succeeds, it will try to download the theme and install it. If this fails, the applet will continue normally with the default Java look and feel.

# 9. Internationalization

The applet can be internationalized using the included properties file.

To support other languages, follow these steps:

1. Uncompress the `jdownload.jar` archive with a ZIP utility.

   ### Note

   .jar files are ZIP files, just rename it to `jdownload.zip`, so you can use your favorite ZIP utility to uncompress the archive.

2. Copy the `messages.properties` file to `messages_xx_YY.properties`, replacing `xx` with a language code and `YY` with a country code.

   ### Note

   The official list of language codes can be found here:

   `http://ftp.ics.uci.edu/pub/ietf/http/related/iso639.txt`

   The official list of country codes can be found here:

   `http://userpage.chemie.fu-berlin.de/diverse/doc/ISO_3166.html`

or `http://www.iso.org/iso/en/prods-services/iso3166ma/index.html`

3. Open the new file (`messages_en_US.properties`) in a text editor.
4. Replace the messages behind the equals symbol ("=") with the translated message.

### Important

Do NOT change the key string on the left side of the equals symbol!

5. Compress all extracted files and your new language file into a new ZIP archive.
6. Rename your ZIP file to `jdownload.jar`
7. Re-sign the jar file with `jarsigner`, a tool included with the Java 2 SDK.

### Note

Before you can use jarsigner, you need to create a self-signed certificate with `keytool`, or import an existing certificate into the keystore with `keytool`.

See Section 10, “Signing” for details.

# 10. Signing

Signing the jar file is necessaray for JDownload to work properly.

Applets are running in a sandbox, a protected environment, which has no access to local file system. JDownload's purpose is to write files to local file system. In order to do the work, it needs access to the local file system. Due to this fact, it must be signed, so it can access the local filesystem.

The signature can be self-signed or trusted by a 3rd party provider. The end-user must accept the certificate (in the security dialog) to let the component access his local file system.

Generating a self-signed test-certificate is easy:

1. Generate a signature with a tool provided by Sun Java SDK called keytool:

```
c:\j2sdk1.4.2\keytool -alias jdownload -genkey
```

2. Enter the required data.
3. Sign the jar file

```
c:\j2sdk1.4.2\jarsigner jdownload jdownload.jar
```

This will sign the application archive (`jdownloa.djar`) with your generated, self-signed (untrusted) certificate.

To get a real code signing certificate, contact companies like Verisign or Thawte.

The root certificate of the company should be integrated into the target browsers, so users will see that the applet is signed by a trusted party.

# 11. Debugging

Debugging the JDownload applet should be done within the IDE. This is for developers only. If you need verbose information on runtime, switch on the debug mode in JDownload and open the Java Console.

The Java Console can be found by right-clicking on the Java icon in the System Tray (bottom right of your Windows desktop).

# Chapter 3. User's Manual

## Table of Contents



This chapter will describe the usage of the JDownload tool from the view of a web user, who wants to download multiple files at once to his local filesystem.

We assume that the JDownload component is already installed on a webserver and correctly integrated into a web application.

# 1. Graphical User Interface

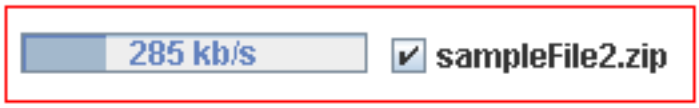
The user interface of JDownload is rather simple. On the left side, you will find an Explorer-like tree with folders and files.

Each folder has an icon to collapse and expand it. Expanding a folder will show it's children - nested subfolders and files. A folder has a name, which directly maps to the folder name which will be created on download.

Each file has a progress bar, a checkbox and a name.

The checkbox is the queueing state of the file. A file can be unchecked and will then not be downloaded. It will just be ignored.

The progress bar shows the percentage done while downloading the file. The current transfer speed (bandwidth in kilobyte per second) is also shown.

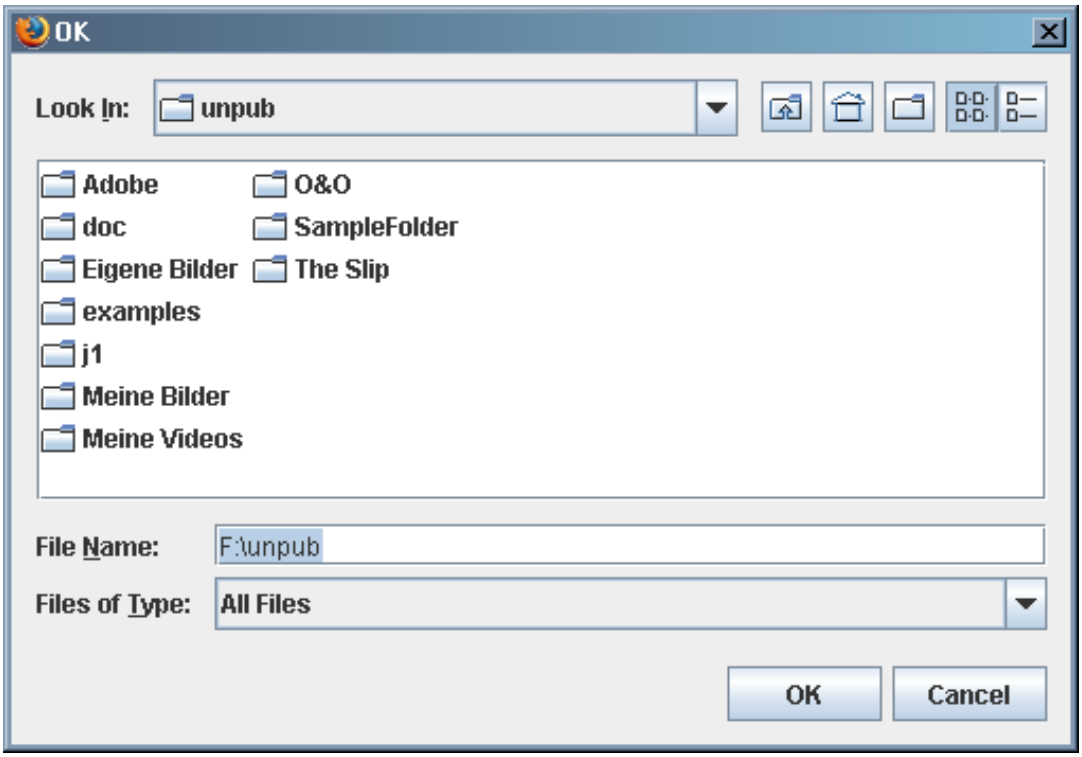



While the tool is downloading all the folders and files, the selection will move on to the current downloading item, which will then be surrounded by a red border.

# 2. Choosing Another Target Folder

The target folder is preset with the My Files folder on Windows systems, or with the home directory on Linux systems.

The user can change the desired download folder (target folder) by clicking on the Browse button.


OK
Look In:
unpub
Adobe
doc
Eigene Bilder
examples
j1
Meine Bilder
Meine Videos
O&O
SampleFolder
The Slip
File Name:
F:\unpub
Files of Type:
All Files
OK
Cancel

# Chapter 4. Appendix

## Table of Contents



Additional information

# 1. Questions And Answers

Why must the jar file be signed?

Normally, an applet neither does have access to system resources, such as the file system, nor is it able to connect to other webservers.

Because these two features are used by JDownload, it must break out of the Java security sandbox. This can only achieved by a signature and the acceptance of the user. The user must actively permit the applet to access his system resources.

The applet does not work on my platform/browser

Although Java is platform independent and should run on every supported platform, there are many complex issues with running applets.

Especially the LiveConnect feature implementation and integration into the different browser types is relatively unstable.

The applet is tested with Internet Explorer and Mozilla Firefox on the Windows platform.

The vendor cannot guarantee that the applet will run on different architectures/platforms/browsers.

# 2. XML Data Specification

This is the XML DTD for the XML Data file used by JDownload.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT download (folder,file)>
<!ELEMENT file (url)>
<!ATTLIST file
  name NMTOKEN #REQUIRED
  uncompress NMTOKEN #IMPLIED
>
<!ELEMENT folder (file|folder)*>
<!ATTLIST folder name NMTOKEN #REQUIRED>
<!ELEMENT url (#PCDATA)>
```

# 3. JavaScript API Event Types

### Table 4.1. Event Types

| No. | Identifier | Description |
|---|---|---|
| 0 | DETACH | The listener has been detached from the notifier. For advanced use only. This event does not have argument data. |
| 1 | ATTACH | The listener has been attached to the notifier. For advanced use only. All events generated by JDownload are now sent to this listener. This event is only sent to the corresponding listener. It does not have argument data. |
| 2 | APPLET_INITIALIZED | The JDownload applet has been initialized and is now ready to be used. The event does not have argument data. |
| 3 | APPLET_WAITING | The JDownload applet is in waiting mode. This event might be deactivated and is for meant debugging purposes. Do not rely on this event being fired. The event does not have argument data. |
| 4 | DOWNLOAD-ITEM_QUEUESTATUS_CHANGED | The user has changed the queueing status of a downloadable item in the Explorer tree. That means he has checked or unchecked the checkbox of a file. Argument 0 contains the DownloadItem object which has been changed, the new status is transferred in argument 1 as boolean value (`true` for enabled/queued and `false` for disabled/unqueued). |
| 5 | TARGETPATH_CHANGED | The user has opened the folder chooser, selected a new target folder and approved to his selection. The old value is in argument 0, the new value is in argument 1. Both values are strings. |
| 10 | DOWNLOAD_STARTING | User has clicked on the Download button, or a JavaScript call to `clickDownload()` has started the download process. The event does not have argument data. |
| 11 | DOWNLOAD_PROGRESS | This event tells that the download process is in progress. Argument 0 is the number of the file currently processed (beginning with 0), argument 1 is the total number of items. |
| 12 | DOWNLOAD_FINISHED | The download process is complete. No argument data avail- |

| No. | Identifier | Description |
|---|---|---|
| | | able. |
| 13 | DOWNLOAD_STOPPED | The download process has been stopped by the user, clicking on the Stop button, or by `clickStop()` being called. No argument data available. |
| 20 | DOWNLOAD-ITEM_STARTING | This event is sent, when a single download item (file) is being started. Argument 0 contains the DownloadItem object being started. |
| 21 | DOWNLOAD-ITEM_PROGRESS | The file is being downloaded, and the event contains information about the progress. Available data arguments:<br>1. The DownloadItem object being processed.<br>2. Percent of the download, as integer value (0-100)<br>3. Current bandwidth in kilobytes per second, as floating point value<br>4. Start time of the item download, in milliseconds since 01.01.1970 00:00:00.00<br>5. Current time of the item download, in milliseconds since 01.01.1970 00:00:00.00 |
| 22 | DOWNLOAD-ITEM_FINISHED | The download of this file has ended. Argument 0 contains the full qualified target filename. |
| 23 | DOWNLOAD-ITEM_STOPPED | The download of this file has been cancelled. No argument data. |
| 30 | DOWNLOAD-ITEM_UNCOMPRESS_STARTING | The archive file which has been completely downloaded is being uncompressed. Argument 0 contains the full qualified filename of the archive, argument 1 contains the target path to where it is being extracted. |
| 31 | DOWNLOAD-ITEM_UNCOMPRESS_PROGRESS | A single entry in the archive file has been uncompressed. The event contains information about the total progress of uncompressing the archive. Update events between two archive entries are not sent, due to performance issues. Argument |

| No. | Identifier | Description |
|---|---|---|
| | | data:<br>1. Filename of the ZIP entry.<br>2. The output file.<br>3. Percentage as integer value<br>4. Current position in the ZIP archive in bytes<br>5. Total length of the uncompressed data in bytes |
| 32 | DOWNLOADITEM_UNCOMPRESS_FINISHED | The archive has been completely uncompressed. Argument 0 contains the filename of the target archive file. |
| 33 | DOWNLOADITEM_UNCOMPRESS_STOPPED | The uncompress has been cancelled. Argument 0 contains the filename of the target archive file. |

# 4. License Agreement

You agree that your use of the SOFTWARE PRODUCT acknowledges that you have read this license, understand it, and agree to be bound by its terms and conditions.

This agreement applies to all software code samples and documentation developed by Mike Haller.

## Important

READ CAREFULLY: This Software Licence Agreement (SLA) is intended to be a legal agreement between The Licensee (as hereinafter defined) and the Authors for the software product identified above which includes computer software and associated media and printed materials and many include “online” or electronic documentation (hereinafter called the “Software”) By installing copying or otherwise using the Software the Licensee agrees to be bound by the terms of this SLA if the terms of this SLA are not agreed do not install, copy or use the Software.

SOFTWARE LICENCE

The Software is protected by copyright laws and other intellectual property laws and rights and the Software is licensed not sold and no property in the Software shall pass as a result of any of the terms of the Licence

1. Definitions

In this Licence the following expression shall have the meanings as attributed to them.

Software shall mean the software product identified above which without prejudice to the generality of the forgoing shall include computer software and associated media and printed materials and may include online or electronic documentation and upgrades.

Licensee shall mean a person individual corporation partnership body corporate or any agency or organisation of whatever nature whether incorporated or not who desires to use and does use the Software.

2. In consideration of the purchase of the Software by the Licensee the Authors hereby grant to the Licensee the non exclusive right to

2.1 install and use the enclosed Software on its computers and other equipment to design develop and test

2.2 modify the sample source code located in the Software to design develop and test the content

2.3 reproduce and distribute the Sample Code in object code form along with any modifications made to the Sample Code PROVIDED ALWAYS THAT the Licensee complies with the Distribution Requirements described in clause 2.4 below and for purposes of this Licence “modifications” shall include enhancements to the functionality of the Sample Code

2.4 Copy and redistribute the Sample Code provided that the Licensee will

2.4.1 distribute the sample code only in conjunction with and as a part of its Application

2.4.2 the Application adds significant and primary functionality to the sample code

2.4.3 not use the Authors name logo or trademarks to market the Licensee Application except with the prior written consent of the Authors

2.4.4 include a valid copyright notice on its Application and

2.4.5 fully and effectively indemnify hold harmless and defend the Authors from and against any claims or demands of whatsoever nature that arise out of or result from the use or distribution of the Licensees Application

2.4.6 the Authors reserve all rights not expressly granted herein

3. All rights title and copyrights in and to the Software and any copies of the Software are owned by the Authors or its suppliers or assigns and the Licensee may not copy the printed materials if any accompanying the Software

4. The Licensee will not nor permit anyone to

4.1 reverse-engineer decompile or disassemble the Software

4.2 rent or lease the Software.

5. The Licensee may permanently transfer all of its rights under this SLA provided it retains no copies and transfers all of the Software (including all component parts the media and printed materials any upgrades this SLA and if applicable the Certificate of Authenticity) and the recipient agrees to the terms of this SLA PROVIDED ALWAYS THAT if the Software is an upgrade any transfer must include all prior versions of the Software

6. Without prejudice to any other rights the Authors may terminate this SLA if the Licensee fails to comply with the terms and conditions of this SLA and in such event the Licensee warrants that all copies of the Software and all of its component parts will be destroyed

7. The Authors expressly disclaim any warranty for the Software and any related documentation are provided as is without warranty of any kind either express or implied including without limitation the implied warranties of merchantability or fitness for a particular purpose and the entire risk arising out of the use or performance of the Software remains with the Licensee

8. In no event shall the Authors its suppliers or assigns be liable for any damages whatsoever (including without limitation damages for loss of business profit business interruption consequential loss or loss of business information or any other pecuniary loss or otherwise) arising out of the use of or inability to use this software even if the Authors have been advised of the possibility of such damages