

# **LinkFixer Web Application User Manual**

## Contents

1. Introduction
  - 1.1. What is it?
  - 1.2. Who can use it?
2. General information
  - 2.1. System overview
  - 2.2. Contact
3. Setting up the LinkFixer Web Application
  - 3.1. Install Xwiki
  - 3.2. Install LinkChecker
  - 3.3. Install Curl
  - 3.4. Import LinkFixer.WebHome.xar in the XWiki
  - 3.5. User Access
4. Using the system
  - 4.1. How are the different links fixed
  - 4.2. Parameter Value Include warnings
  - 4.3. Output file
  - 4.4. Recursion depth
  - 4.5. Number of threads
  - 4.6. Number of noFollow links
  - 4.7. Linkchecker result file
  - 4.8. Domain to fix
  - 4.9. Submit and abort
  - 4.10. LinkFix

## **Introduction**

This user guide documents the use of the LinkFixer Web Application.

### **What is it?**

The LinkFixer Web Application implements automated fixing of broken links for XWiki CMS, fixing content and storing the necessary site documents.

### **Who can use it?**

The LinkFixer Web Application is made as such that it can be used by anybody with average computer skills. No special system administration and programming skills are required.

## **General information**

### **System overview**

1. Web interface for user friendly input setting configuration.
2. Launch LinkChecker to view the broken links
3. Launch LinkFixer to automatically fix 10 different types of broken links found by LinkChecker
4. User guide that explains the usage/setup of the project

The LinkFixer Web Application allows input of data through manual input into the entry grids.

### **Contact**

Requests for access to, inquiries on the use of the software, and comments on the design and functionalities of the application should be written in <https://github.com/EdmundsOzolins/LinkFixer/issues> .

## Setting up the LinkFixer Web Application

This chapter describes the different steps you should take to get the application up and running.

### Install Xwiki

All installations are available on the following URL:  
<https://www.xwiki.org/xwiki/bin/view/Documentation/AdminGuide/Installation/>.

### Install LinkChecker

All installations are available on the following URL:  
<https://github.com/wummel/linkchecker/blob/master/doc/install.txt>.

### Install Curl

All installations are available on the following URL: [https://www.luminanetworks.com/docs-lsc-610/Topics/SDN\\_Controller\\_Software\\_Installation\\_Guide/Appendix/Installing\\_cURL\\_for\\_Ubuntu\\_1.html](https://www.luminanetworks.com/docs-lsc-610/Topics/SDN_Controller_Software_Installation_Guide/Appendix/Installing_cURL_for_Ubuntu_1.html)  
<https://curl.haxx.se/download.html>.

### Import LinkFixer.WebHome.xar in the XWiki

Download LinkFixer.WebHome.xar file from the following URL:  
<https://github.com/EdmundsOzolins/LinkFixer/blob/master/LinkFixer.WebHome.xar>.  
Import LinkFixer.WebHome.xar in the Xwiki. All import instructions are available on the following URL: <https://www.xwiki.org/xwiki/bin/view/Documentation/AdminGuide/ImportExport>.

### User Access

Ensure that the user has edit rights.

Use [your\_site]/xwiki/bin/view/LinkFixer/GUI for fixing broken links

## Using the system

### How are the different links fixed.

The table below explains how are the different links fixed:

| Name                            | Description                                     | Xwiki syntax              | Fix                       |
|---------------------------------|---|---------------------------|---------------------------|
| Implicit URL                    | Converts to url automatically                   | url                       | Delete                    |
| Explicit URL                    | As implicit, but with extra [[]]                | [[url]]                   | Delete                    |
| URL with label                  | Link which is hidden by custom label            | [[label>>url]]            | Change to label name      |
| Link to email address           | Like URL with label, but with mailto:           | [[email>>mailto:email]]   | Change to email           |
| Link to image                   | A link to some image                            | [[image:url]]             | Delete                    |
| Link on an image                | Clicking this image results in following a link | [[image:url1>>url2]]      | Delete                    |
| Attachment                      | Attaches some file                              | [[attach:file]]           | Replace with name of file |
| Link that opens in a new window | Clicking on the link will open it in a new page | [[url  target="_blank"]]  | Delete                    |
| An url link                     | url   | [[url:url]]               | Delete                    |
| An ftp link                     | A link to a file                                | {{some file="location"}}} | Replace with name         |

The picture below shows how LinFixer Web Application looks like:

LinkFixer

Last modified by Administrator on 2019/08/29 12:58

LinkFixer is a web application that fixes broken links of HTML documents and websites. It relies on results produced by LinkChecker command line client (written by Bastian Kleineidam). User must have LinkChecker installed in case a LinkChecker output file has not been produced prior to the usage of LinkFixer. If such file exists, it is sufficient to have the LinkChecker output file somewhere in the user's filesystem. Curl command line tool must be installed to use LinkFixer.

### Launch LinkChecker

**INCLUDE WARNINGS**  
Choose whether warnings should be logged.

Options

**OUTPUT FILE**  
Specify output type.

none

**NUMBER OF NOFOLLOW LINKS**  
Check but do not recurse into URLs matching the given regular expression. Provide RegEx in the generated fields.

0

**NUMBER OF IGNOREURLS**  
URLs matching the given regular expression will be ignored and not checked. Provide RegEx in the generated fields.

0

**RECURSION DEPTH**  
Check recursively all links up to given depth. A negative depth will enable infinite recursion.

-1

**NUMBER OF THREADS**  
Generate no more than *the* given number of threads. Default number is 100.

100

**LINKCHECKER RESULT FILE**  
FILENAME of an HTML file containing a list of broken links detected by LinkChecker. FILENAME is used by LinkFixer to fix the broken links.

**DOMAIN OR HTML TO FIX**  
Domain or HTML file that require fixing.

Submit Abort

### Launch LinkFixer

**USERNAME**  
Username for HTTP and FTP authorization

**PASSWORD**  
Password for HTTP and FTP authorization

**LINKCHECKER RESULT FILE**  
FILENAME of an HTML file containing a list of broken links detected by LinkChecker. FILENAME is used by LinkFixer to fix the broken links. Default: FILENAME produced in LinkChecker run.

LinkFix Reset

Tags: [+]

Created by Administrator on 2019/08/27 09:44

## Include warnings

**Yes** - include warnings.

**No** - only errors.

## Output file

Valid file output types are **text**, **html**, **sql**, **csv**, **gml**, **dot**, **xml**, **sitemap**, **none** or **blacklist**. Default is no file output. Note that you can suppress all console output with the option **none**.

**text** - Standard text logger, logging URLs in keyword: argument fashion.

**html** - Log URLs in keyword: argument fashion, formatted as HTML. Additionally has links to the referenced pages. Invalid URLs have HTML and CSS syntax check links appended.

**csv** - Log check result in CSV format with one URL per line.

**gml** - Log parent-child relations between linked URLs as a GML sitemap graph.

**dot** - Log parent-child relations between linked URLs as a DOT sitemap graph.

**gxml** - Log check result as a GraphXML sitemap graph.

**xml** - Log check result as machine-readable XML.

**sitemap** - Log check result as an XML sitemap whose protocol is documented at <http://www.sitemaps.org/protocol.html>.

**sql** - Log check result as SQL script with INSERT commands. An example script to create the initial SQL table is included as create.sql.

**blacklist** - Suitable for cron jobs. Logs the check result into a file `~/.linkchecker/blacklist` which only contains entries with invalid URLs and the number of times they have failed.

**none** - Logs nothing. Suitable for debugging or checking the exit code.

### **Number of nofollow links**

Check but do not recurse into URLs matching the given regular expression.

### **Number of ignoreurls**

URLs matching the given regular expression will be ignored and not checked. This option can be given multiple times.

### **Recursion depth**

Check recursively all links up to given depth. A negative depth will enable infinite recursion. The maximum recursion level must not be exceeded.

### **Number of threads**

Generate no more than the given number of threads. Default number of threads is 100. To disable threading specify a non-positive number.

### **Linkchecker result file**

Create a new file and copy file path into the “Linkchecker result file”. The file type should be the same as “Output file”.

### **Domain or html to fix**

Link of domain or html to fix.

### **Submit , reset and abort**

Click on “Submit” button and wait for results to appear.

“Abort” button – abort the process.

“Reset” button – reset form fields.

### **Username, password**

User must have edit rights.

### **LinkFix**

Click on “LinkFix” button to fix broken links.