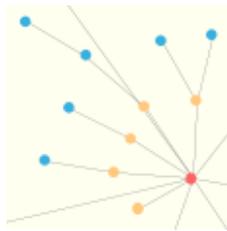


User manual for Jwire 1.0  
septembre, 2010

## User manual for Jwire 1.0

Luis Alberto García Hernández





Information about the author:

Luis Alberto García Hernández  
Informatic engineer  
luisalbertogh@hotmail.com  
<http://it.luisalbertogh.net>

This document can be freely distributed.  
(c) 2010 Luis Alberto García Hernández



## Abstract

**WIRE** is a *crawler* or web spider developed by a team from the University of Chile under GPL license, that is, of public use. The crawler is able to crawl and extract huge amount of information from Internet, accessing several web sites. It is able to crawl a whole domain of a given country if it has the necessary computational resources and time.

This crawler is used by professors and researchers from the University of Salamanca in the realization of their works, within different fields of Informatics, like those ones related to Web Mining and Cybermetrics. Due to the complexity of the operations performed by the crawler and the lack of graphical user interfaces, the use of WIRE can become annoying and complex for non-advanced users. Also, in the case of the crawlers in general, and with WIRE in particular, the amount of information retrieved from the crawled networks is, in most of the cases, very important. To analyze such amount of data can be a complex work without the necessary techniques and tools.

The goal of this project is to analyze the different visualization techniques that can be applied to the analysis of the data obtained by a crawler, in this case the crawler WIRE, and to develop a set of visualization tools that implement those techniques in order to verify and validate their use. This set of tools will be developed as a graphical environment for WIRE, named **Jwire**. Thus, the original project title named “**WIRE, a multifunction crawler**” has been replaced by the current one. Adding to WIRE a new graphical user interface from which it could be possible to make use those visualization tools, together with the whole set of WIRE functionalities, but in a simpler and quicker way, making WIRE more accessible to any kind of user, automatizing tasks and in summary adding additional values to the already existent application.

This document presents a brief graphical description of the functionalities of Jwire in its version 1.0. For more information about the operations of Jwire, consult the technical manual.

# Contents

<b>List of Figures</b>	<b>iii</b>
<b>1 Requirements</b>	<b>1</b>
<b>2 Installation and execution</b>	<b>1</b>
<b>3 Configuration module</b>	<b>2</b>
3.1 Edit wire.conf . . . . .	2
3.2 Edit urls file . . . . .	3
<b>4 Execution module</b>	<b>5</b>
4.1 Execution of WIRE commands . . . . .	5
4.1.1 Create several execution lists . . . . .	6
4.1.2 Visualizing and exporting the standard output . . . . .	6
4.1.3 Controls bar . . . . .	7
4.2 Managing WIRE reports . . . . .	8
<b>5 Visualization module</b>	<b>9</b>
5.1 Force-directed web graph . . . . .	11
5.2 Multilayer graphs . . . . .	13
5.3 Tools for graphs . . . . .	17

## List of Figures

1	Editing wire.conf . . . . .	2
2	Commands toolbar . . . . .	3
3	Properties of wire.conf . . . . .	3
4	Editing the urls file . . . . .	4
5	Commands toolbar . . . . .	4
6	Urls table . . . . .	5
7	Execution module . . . . .	5
8	Command execution . . . . .	6
9	Several lists at the same time . . . . .	7
10	Standard output and error . . . . .	7
11	Controls bar . . . . .	7
12	Configuring applications . . . . .	8
13	Accessing the files . . . . .	8
14	Visaulization module . . . . .	9
15	Visualization toolbar . . . . .	11
16	Visualizind data in Jwire . . . . .	12
17	Force-directed web graph . . . . .	13
18	Information about the graph . . . . .	13
19	Hierarchical aggregation of the multilayer graph . . . . .	14
20	Level 0 . . . . .	14
21	Level 1 . . . . .	15
22	Level 2 . . . . .	16
23	Toolbar for graphs . . . . .	17
24	Radial graph . . . . .	18
25	Treemap . . . . .	19
26	Force panel and nodes searcher . . . . .	19
27	Data about the nodes . . . . .	20



## 1 Requirements

**Jwire** is implemented in standard **Java** with the SDK from Sun, version 6 update 18. Thus, Jwire requires the Java JRE, that has to be previously installed in the system on which is gonna be executed, in its version 5 or upper. It is recommended version 6 update 10 or upper, in order to make use of all its features.

## 2 Installation and execution

Unzip or untar the archive Jwire (**zip** or **tar**) in the folder from where it is gonna be launched. The Jwire archive contains all the files and folders required for Jwire. Jwire will consider the current directory as the root directory in which store all the information needed for its right performance.

To execute Jwire, launch the execution script associated to the corresponding operative system in which is installed (**.sh** for Unix/Linux or **.bat** for Windows).

## 3 Configuration module

### 3.1 Edit wire.conf

1. Select the **Configuration** tab
2. Do click on the button with label **wire.conf**

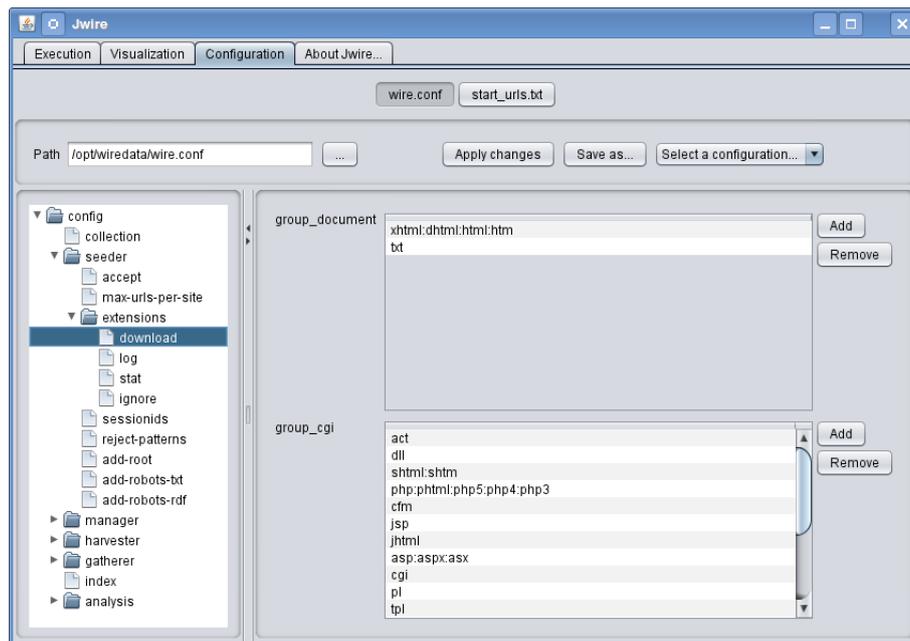


Figure 1: Editing wire.conf

- Use the path field to select a new configuration file
- Use **Apply changes** to save the changes
- Use **Save as...** to save the changes to a file without applying the changes
- Use the drop-down **Select a configuration...** to load the data stored in a file for later modification

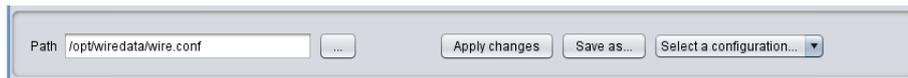


Figure 2: Commands toolbar

- Select the nodes in the tree that correspond each of the fields to edit in wire.conf
- On the right side, it appears the values that can be edited for the selected field. There are different types of fields. Each field has different controls, some of them are numeric values, some others are tables, others are text fields, etc.



Figure 3: Properties of wire.conf

## 3.2 Edit urls file

1. Select the tab **Configuration**
2. Do click on the button with label `start_urls.txt`

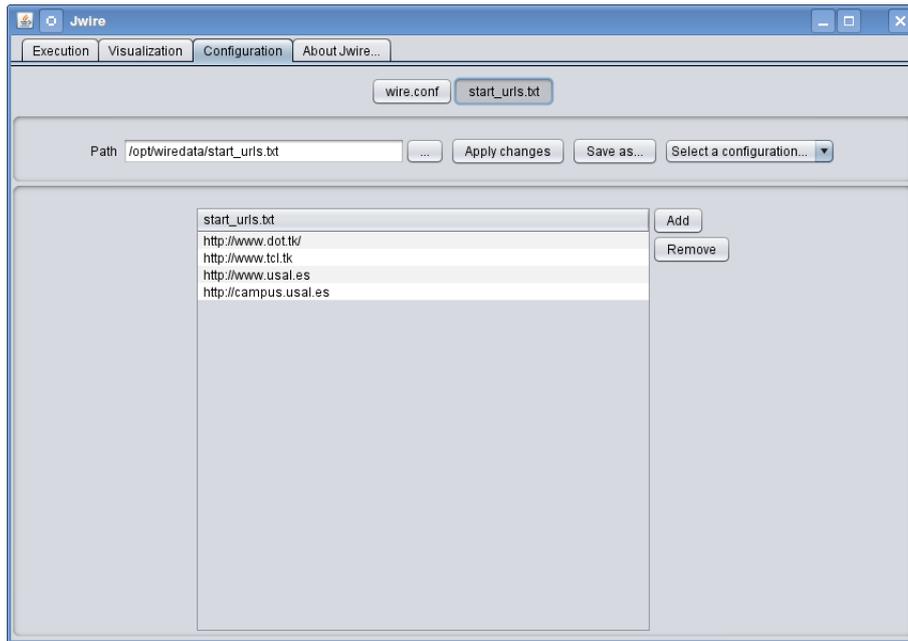


Figure 4: Editing the urls file

- Use the path field to select a new urls file
- Use **Apply changes** to save the changes done
- Use **Save as...** to store the changes in a file without applying them to the urls file
- Use the drop-down **Select a configuration...** to load the data stored in a file for its later modification



Figure 5: Commands toolbar

- Use the table to add, delete and edit **urls** for their use with Jwire

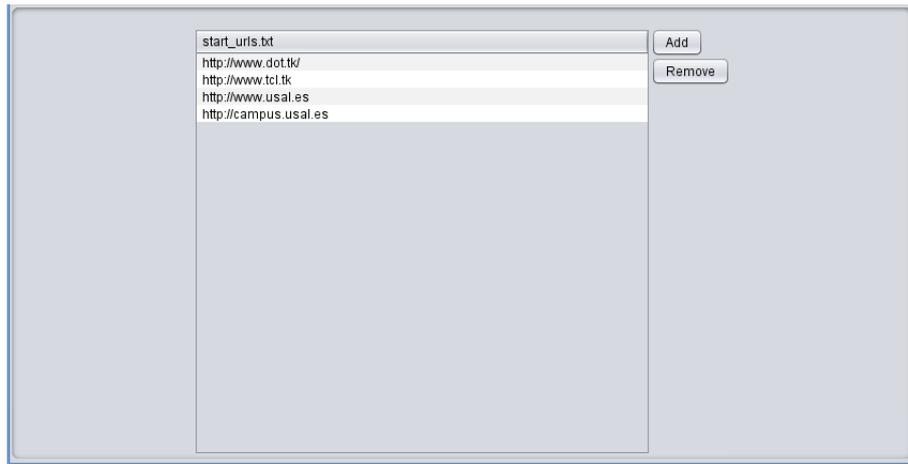


Figure 6: Urls table

## 4 Execution module

### 4.1 Execution of WIRE commands

After selecting the **Execution** tab...

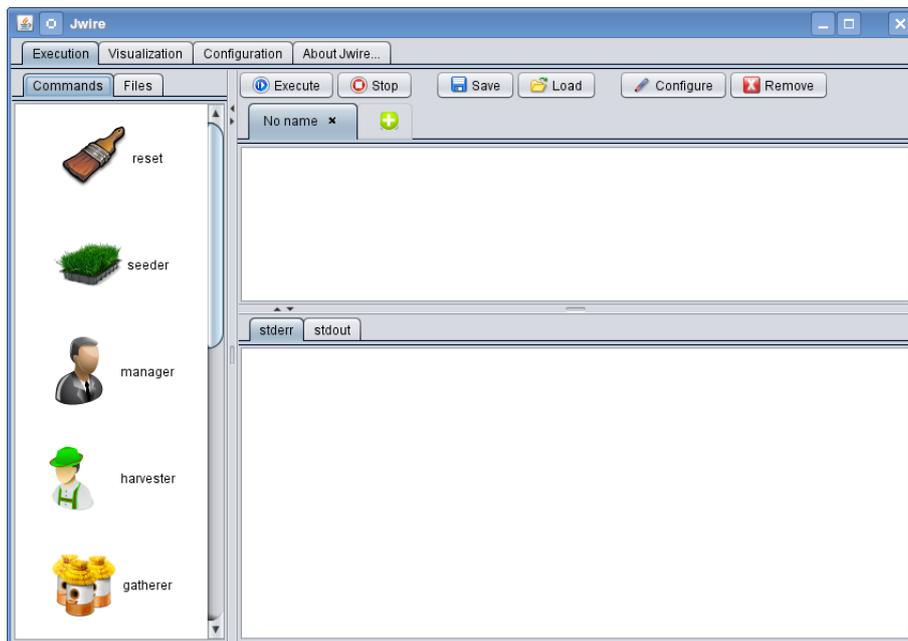


Figure 7: Execution module

1. Select the WIRE command from the commands list
2. Drag the selected commands to the execution list
3. Optionally, configure each of the execution parameters of each command, using the **Configurar** button from the control bar

- Execute the selected list. The commands from the list will be executed in order, from left to right. The command in execution appears with red background while it is being executed.
- In the lower part of the window, it appears the standard output and error of the execution of the commands.

For more information about the control bar in the execution module, see section 4.1.3.

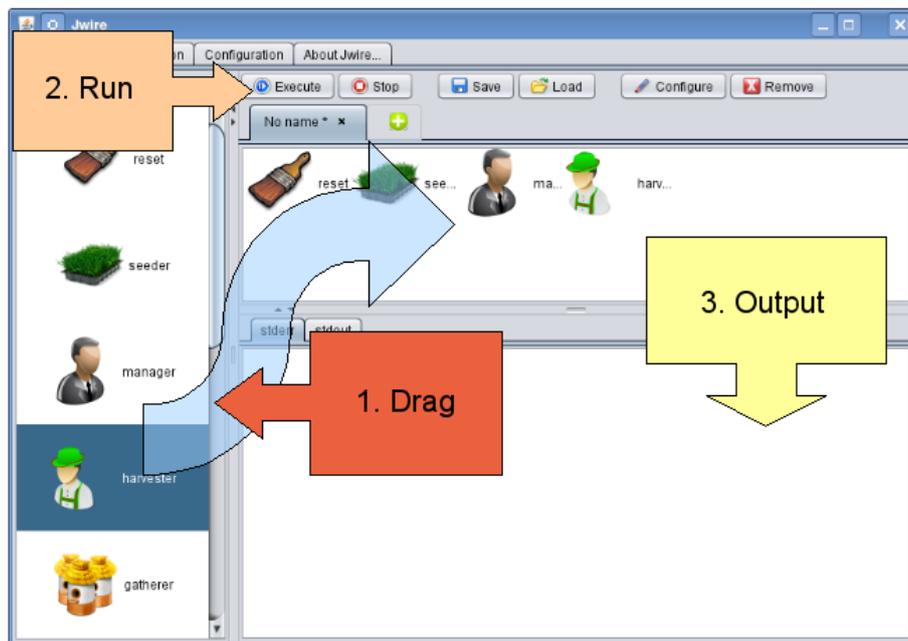


Figure 8: Command execution

### 4.1.1 Create several execution lists

It is possible to handle several execution lists at the same time, creating new tabs in the upper part of the central canvas. It is enough by clicking on the green circle with the white cross that is shown next to the existent tabs. Each time that a new tab is created, the title **No name** is assigned by default. This can be changed when saving the execution lists, as it is mentioned below.

### 4.1.2 Visualizing and exporting the standard output

There are two different panels associated to each execution list, that show the corresponding standard output and error of each executed command from the list. The content of these panels can be exported to a text file by **clicking with the right button of the mouse** to show a **pop-up menu**, and then choosing the option **Export** and specifying the **destination file**.

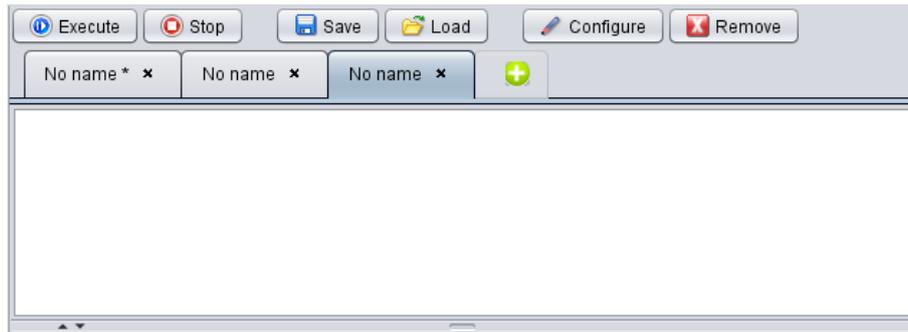


Figure 9: Several lists at the same time

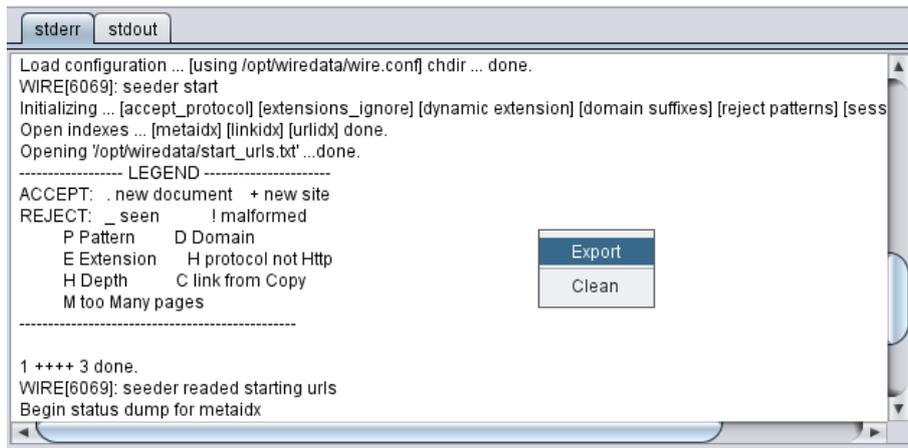


Figure 10: Standard output and error

### 4.1.3 Controls bar

There are different controls inside the bar situated in the upper part of the central execution panel of Jwire. Those controls have the following functions:

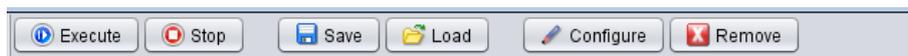


Figure 11: Controls bar

- **Execute.** Execute the command list
- **Stop.** If possible, it stops the execution of the command list
- **Save.** Save the command list into a file, for later reutilisation
- **Load.** Load into Jwire a command list from a file, that has been previously saved
- **Configure.** It allows configuring each WIRE command parameters, depending on the needs for each command
- **Delete.** Delete from the command list the selected command

## 4.2 Managing WIRE reports

From Jwire it is possible to open the reports generated by the WIRE commands. Jwire makes use of the existent applications on the operative system on which it is installed in order to open those files. Thus, before making use of this functionality, it is necessary to configure Jwire to indicate it with what applications the different files will be open.

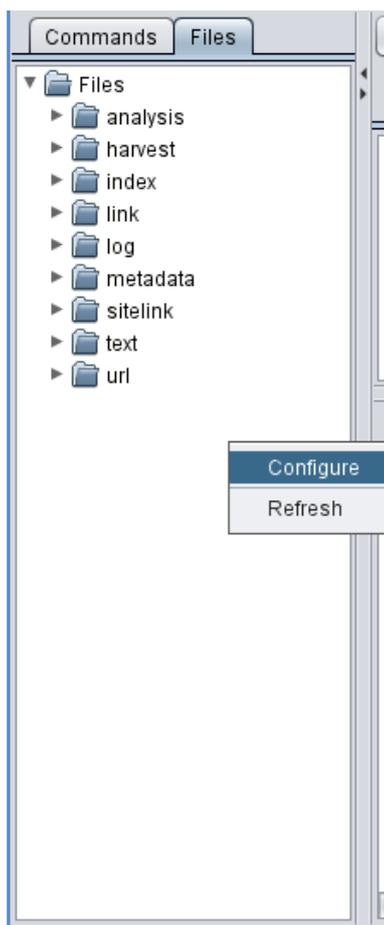


Figure 12: Configuring applications

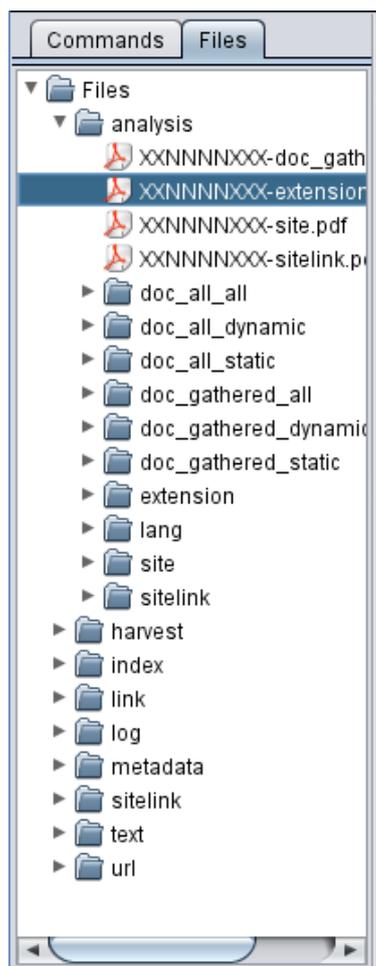


Figure 13: Accessing the files

For this:

1. From the execution tab, select the **Files** tab from the left panel
2. A hierarchical tree is shown with the structure of the files and folders created by WIRE in order to store the generated reports. Clicking on the right button of the mouse on the panel, it is shown a pop-up menu. Choosing the option **Configure**, it is open a dialog box that contains a table. This table stores in two columns the paths to the applications to use and the extensions of the files associated to those applications. Using the dialog box buttons or clicking on the table rows it is possible to edit, add or delete any association. If the value

\* is indicated as extension, the associated application will open any other file that won't have any other association set on the table (wildcard).

Once Jwire is configured, the files generated by WIRE can be open in the following way:

1. From the execution tab, select the **Files** tab from the left panel in the window
2. A hierarchical tree is shown, with the structure of the folders and files created by WIRE to store the generated reports. Browse inside this structure until finding the searching file and double-click on it to open it in a separated application

Jwire searches for the generated files by WIRE in the default location selected by WIRE to store them, and comes from the folder selected by WIRE to store the data. To configure the path of this folder, it is necessary to modify the **wire.conf** file and the WIRE **environment variable**.

Finally, the content of the reports directory tree can be refreshed, making use of the **Refresh** option of the pop-up menu from the **Files** tab.

## 5 Visualization module

The Jwire visualization and visual analysis tools is available by activating the **Visualización** tab.

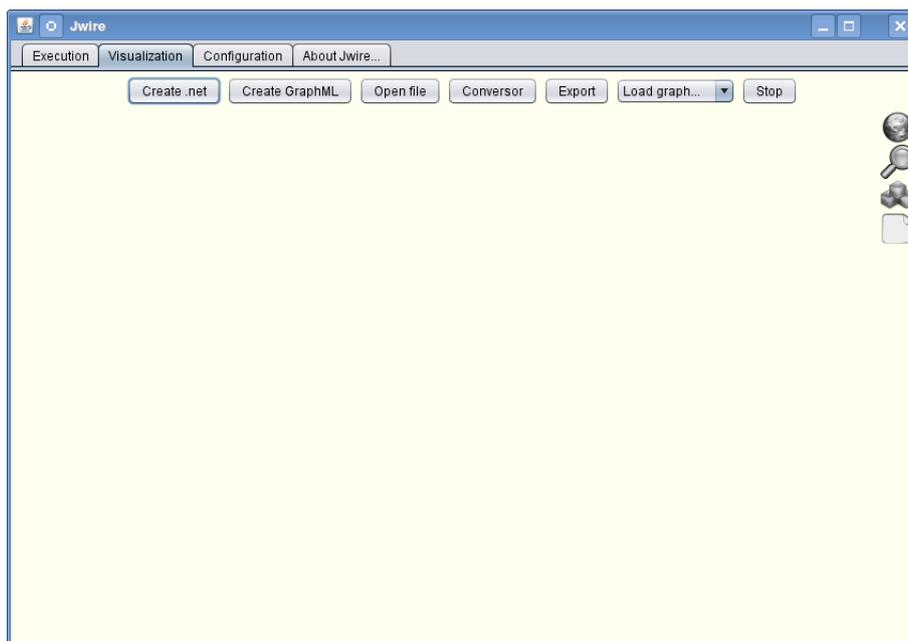


Figure 14: Visaulization module

Jwire offers to WIRE a set of visualization tools that implement most of the standard techniques for visualization and to analyze the data extracted from a network, in particular with the use of **web graphs** and **treemaps**. The visualization module makes use of the WIRE commands used to format and generate the extracted data, and afterwards it uses the recommended visualization techniques to show those data on the screen in the form of web graphs and other useful visualizations when analyzing the obtained data.

It does exist a tool bar in the form of a sequence of buttons, that offers different functionalities. These are the following:



Figure 15: Visualization toolbar

- **Create .net.** Generate a file in .net (Pajek) format with the data extracted in the last WIRE execution. This is useful when visualizaing or manipulating those data with applications like Pajek or others that only work with the .net format
- **Create GraphML.** Generate a XML file with the GraphML schema for web graphs. This is a necessary step in order to visualize in Jwire the data generated by WIRE
- **See files.** Open in a sperated windows the selected file and show it in text mode
- **Conversor.** Convert files from .net to GraphML and viceversa
- **Export.** Export the generated graph in the area of visualization to an image
- **Load graph...** Once the GraphML file has been generated, it can be open as a web graph by selecting it. There are two visualization modes, the **force-directed graph**, that shows the totality of the data generated in a web graph directed by an artificial forces system, or the **multilayer graph**, that shows the data distributed in a hierarchical structure that composes a graph of 3 different levels
- **Stop.** Stop or restart the web graph movement

The way to visualize the extracted data is the following:

1. Generate the XML file with the GraphML format, clicking on the corresponding button of the tool bar. The file has to have the **.xml** extension
2. Load the file with the graph. Use the drop-down with the tool bar, selecting firat the type of graph that it is visualized and afterwards the corresponding XML file

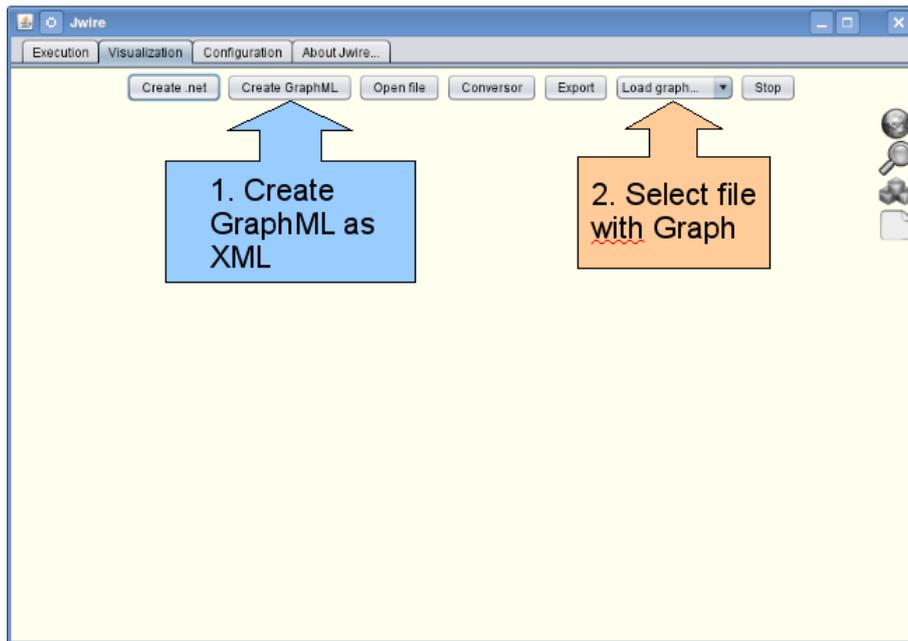


Figure 16: Visualizing data in Jwire

## 5.1 Force-directed web graph

The following is an example of the visualization of data generated by WIRE through a force-directed graph.

By default, the aggregations are shown inside the graph, through the colored spots that group the nodes that belong to the same web site. That is, each colored spot is a single web site, like **www.something.com**.

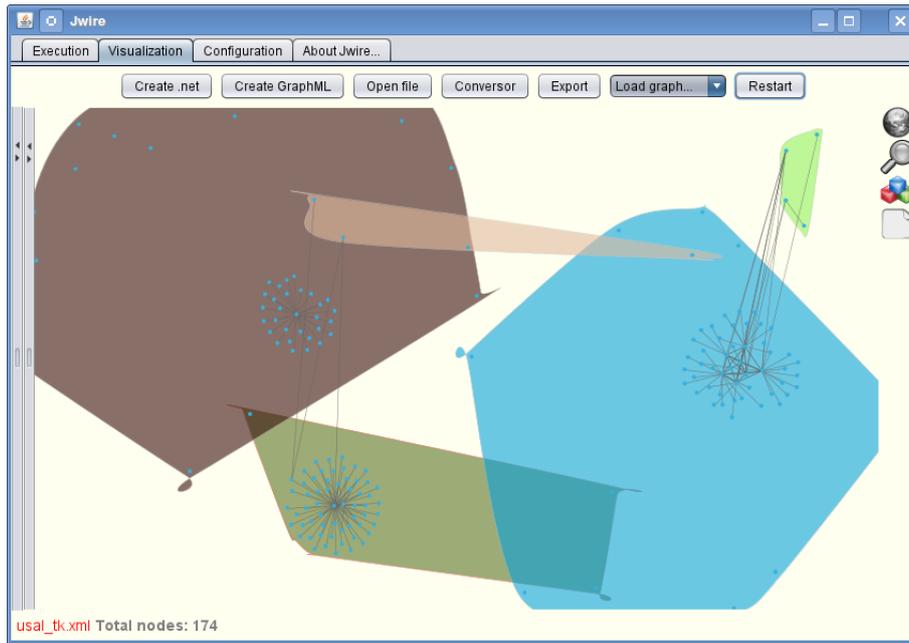


Figure 17: Force-directed web graph

In the lower part of the visualization area, it is shown the information about the graph and the nodes and links that are selected by hovering the pointer of the mouse on them. Normally, it is shown the name of the open file, the total number of nodes, the node or aggregation name selected and the number of nodes of the aggregation.



Figure 18: Information about the graph

## 5.2 Multilayer graphs

El grafo multicapa es un conjunto de grafos web que representan diferentes niveles dentro de una estructura jerárquica de contención. Los datos generados por WIRE se distribuyen en un único nivel, al cual se accede desde otros niveles superiores que se crean de forma artificial y que representan las agrupaciones de esos datos de forma distribuida. El siguiente diagrama representa el mecanismo de construcción de los niveles del grafo multicapa.

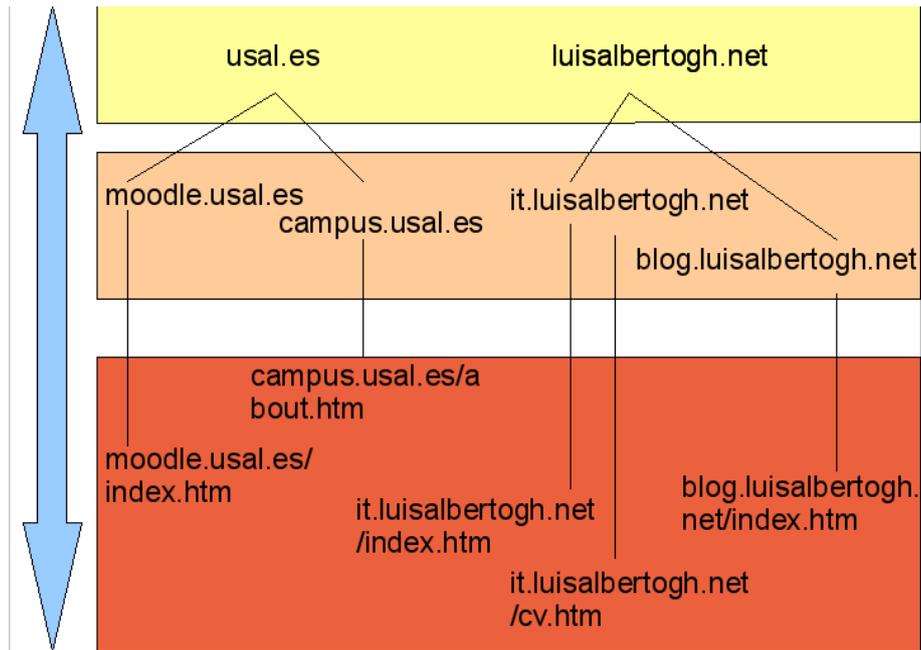


Figure 19: Hierarchical aggregation of the multilayer graph

The nodes of the lowest level are extracted by WIRE. Jwire shows by default the uppest level, and it is possible to access the lower levels by double clicking on the node that represents the aggregation to explore.

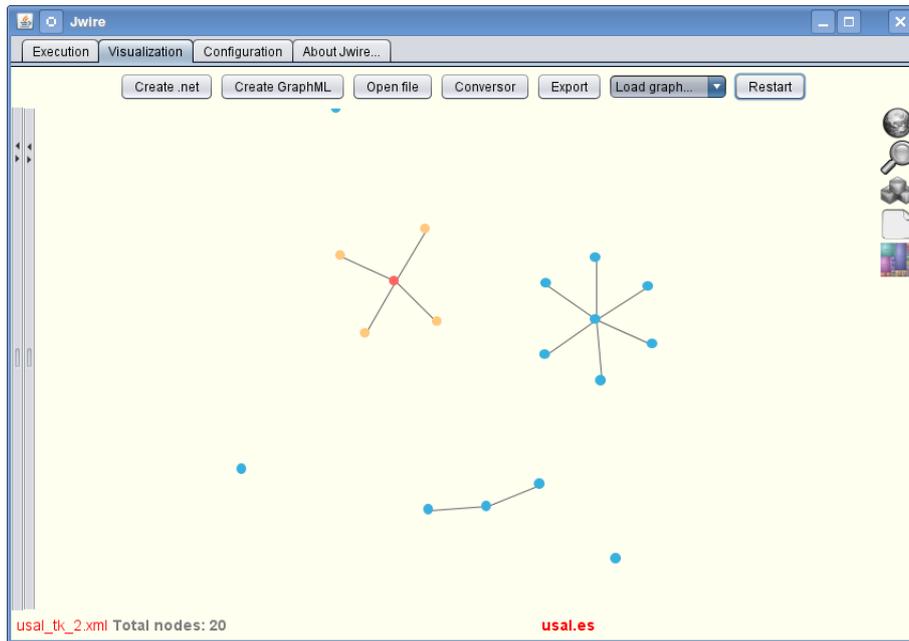


Figure 20: Level 0

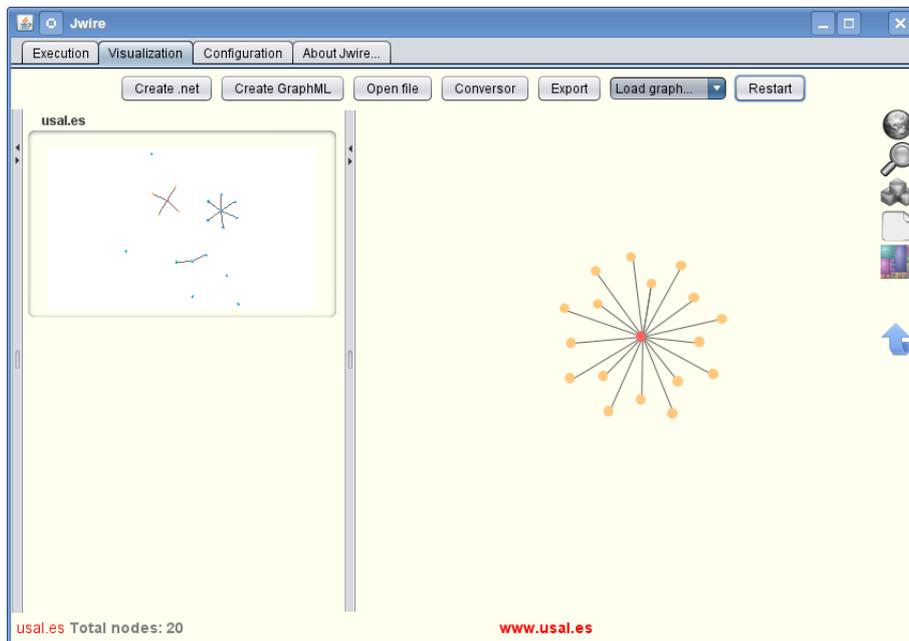


Figure 21: Level 1

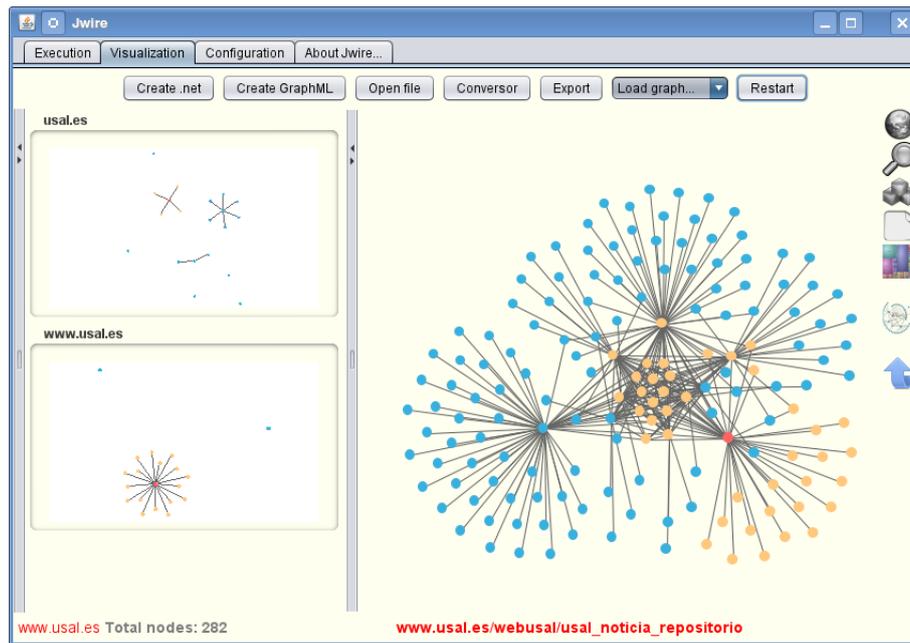


Figure 22: Level 2

As it can be seen in levels 1 and 2, it is always shown the previous level in a shrunk view on the left side, so it is always kept visible from where the view is coming, and by double clicking on that view it is possible to access again to the corresponding level.

### 5.3 Tools for graphs

Either for the force-directed web graph or for the multilayer graph, there is a set of tools implemented that allow the manipulation and visualization of the different components of the graphs. These tools implement most of the standard techniques for graph visualization and are a recommendation when visualizing the graphs in the best way.

Most of these tools are accessible from the task bar that appears on the right part of the visualization area:



Figure 23: Toolbar for graphs

This tool bar is not always the same. The available options are different for each type of graph. The ones shown in the image are all of the existing ones, and represent the following functions, from top to bottom:

1. **Global view.** It shows on the graph a reduced area that contains the global view of the graph, and a rectangular area that is able to spot a determined zone of the graph that is shown amplified in the central area of the window
2. **Magnifying glass.** Using the mouse pointer, it can be magnified a concrete zone of the graph, while the rest can be seen in a reduced way, as it can be seen when using a magnifying glass
3. **Aggregations.** This option is used to show or hide the aggregations. In the case of the force-directed web graph, is activated by default, and shows the web sites with colored spots. In the case of a multilayer graph, is deactivated by default, but it can be activated by selecting the type of aggregation to show, either on the second level domains or on the first level domain. In this case, the aggregations are shown by coloring with the same color the nodes

that belong to the same group **Tooltips**. Show the name of the nodes if the mouse pointer is hovering on the nodes. Deactivated by default

4. **Treemap**. Only for multilayer graph. Show in a separated window a treemap of the current graph
5. **Radial graph**. Only for the last level of the multilayer graph. It shows in a separated window the radial graph of the current graph
6. **Up one level**. Only for multilayer graph, it is used to go up one level from the previous one

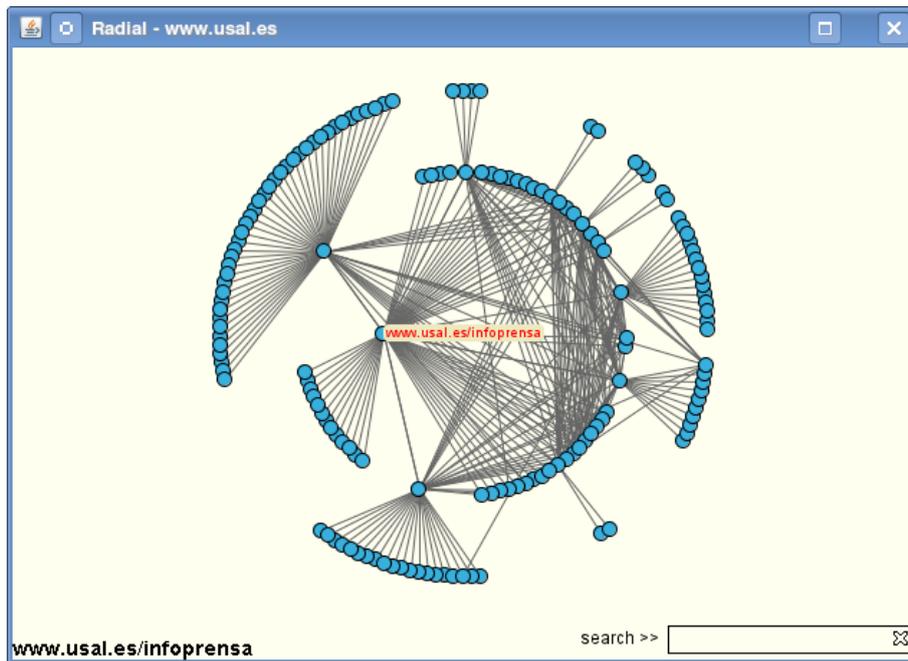


Figure 24: Radial graph

There is a panel that allows modifying the properties of the system of forces that controls the disposal of the graphs, and a searcher to search nodes by name. The panel is hidden by default, but it can be shown by clicking or dragging on the black arrows in the left side of the window.

Finally, certain features have been added to the web graph in order to ease its visual analysis. The selected nodes with the mouse pointer are colored in **red** automatically, and in **yellow** the nodes directly linked with them. When clicking on a node while pressing the **SHIFT** key, it is shown a small window with some data about the connectivity of the node inside the graph (grades, neighbourghs, etc.). If it is clicked the node while pressing the **Control** key, the default web browser of the system will be open and it will try to access the web address represented by the node.

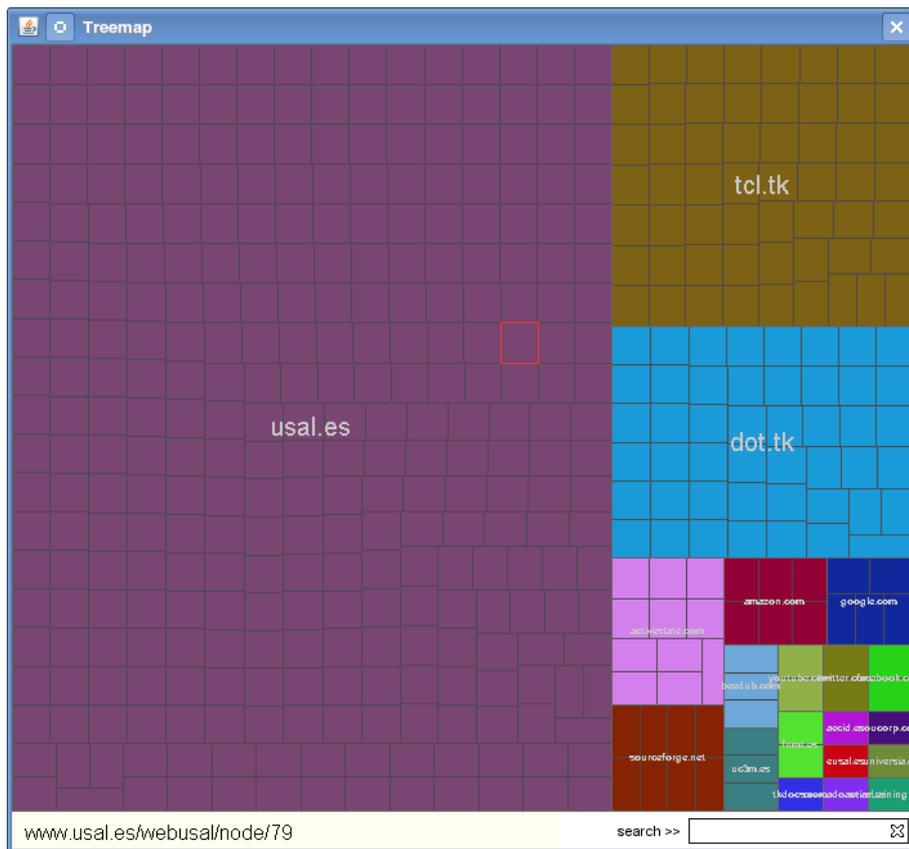


Figure 25: Treemap

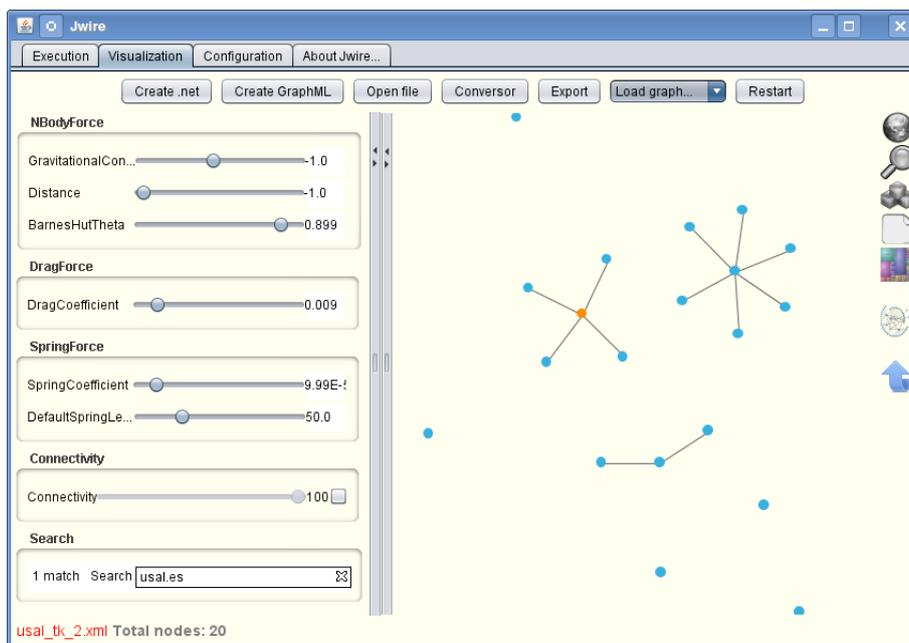


Figure 26: Force panel and nodes searcher

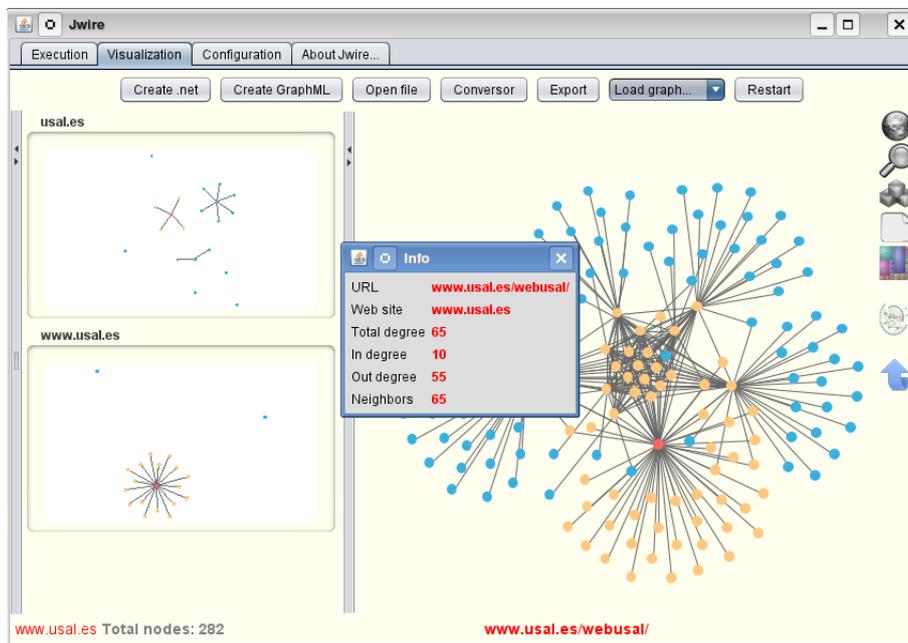


Figure 27: Data about the nodes