OutWit Hub Light, Pro, Expert & Enterprise Editions v8.x

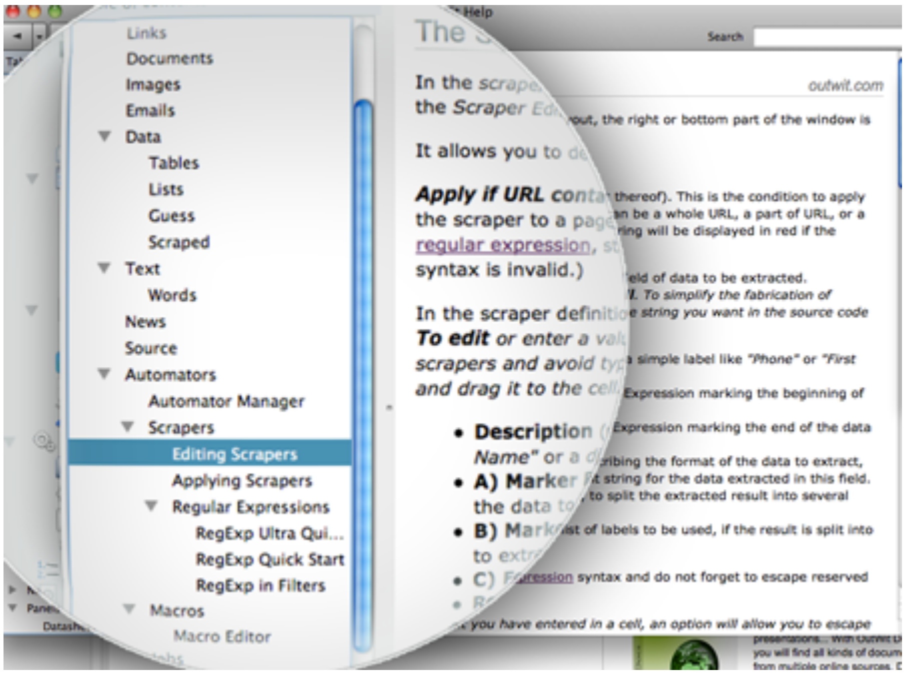
Welcome to OutWit Hub Help Center

***Important Note**: This documentation covers OutWit Hub's features, regardless of your license level and the actual presence of these features in the version you are using. Many views and functions are only available in OutWit Hub Pro, Expert or Enterprise editions. If you are using OutWit Hub Light, please consider this online help as a preview of the advanced features.*

Do not forget to read the Frequenly Asked Questions section in this Help Center.

You can also have a look at the list of recent additions and corrections as well as the main features of each edition

.



*To display help on OutWit Hub in this window, click on one of the topics in the side panel. **Click on the plus sign or triangle near a topic to see its sub-topics**.*

You can also use the search box to locate a specific topic in the OutWit help.

The complete online help is accessible directly from the OutWit Hub application through the Help menu or by clicking on the question mark icon in every view of the program's interface.

For a quick overview on how to navigate on the OutWit Hub interface, click here.

If you cannot find what you are looking for in this integrated OutWit Help, please check the support section in the OutWit Web Site.

Frequently Asked Questions

Here are answers to the most frequent questions. *[updated:2/23/18]*

General

What is OutWit Hub and when should I use it?
When you are looking for something on the Web, search engines give you lists of links to the answers. The purpose of OutWit Hub is to actually go retrieve the answers for you and save them on your disk as data files, Excel tables, lists of email addresses, collections of documents, images...

If your question has one simple answer, it will be at the top of Wikipedia or Google results and you don't need OutWit for that. When you know, however, that it would take you 20, 50, 500 clicks to get what you want, then odds are you do need OutWit Hub:

The Hub is an all-in-one application for extracting and organizing data, images, documents from online and local sources. It offers a wealth of data recognition, autonomous exploration, extraction and export features to simplify Web research. OutWit Hub exists both as an add-on for Firefox (up to FF version 42) and as a standalone application for Windows, Mac OS, and Linux.

OK, I have downloaded OutWit Hub and I am running it. Now what?
We have an open list of 1,728 first things you can do with the application but we believe the best first thing is to run the built-in tutorials from the Help menu (Help>Tutorials).

Automatic Exploration

I want OutWit Hub to browse through a series of result pages but the 'Next in Series' and 'Browse' buttons are disabled. How come?
When opening a Web page, OutWit analyzes the source code and tries to understand as many things as possible about the page. The first thing it does is to find navigation links (next, previous…) and, when it does, the *'Next in Series'* arrow and *'Browse'* double arrow become active. If they are inactive, it is because OutWit did not find any additional pages. There are many workarounds to do the scrape without having to click on all links manually: depending on the cases, the best alternative solutions are using the *Dig* function (with advanced settings in the Pro, Expert or Enterprise editions), *generating the URLs* to explore, making a 'self-navigating' scraper with the *#nextPage# directive* or, finally, grabging the URLs you want to scrape, putting them in a directory of queries and using this directory to do a new automatic exploration. *(Note that for the latter, it is also possible to grab the links to the Catch in one macro and address the column of the Catch by the name you gave it in a second macro, by typing 'catch/yourColumnName' in the Start Page textbox.)*

Some links are not working in the Standalone version of the Hub. What should I do?
These are links for which target=blank was specified in the source code. OutWit Hub cannot open separate popup windows but you can open them within the Hub. For this, check the "Open popup links in the application window" preference (Tools>Preferences>General).

Auto-Browse Function keeps stalling. What can I do?
There can be several reasons for this but the most likely is that some pages do not load fast enough for the timout set in the preferences. In this case the program doesn't have time to find the link to the next page and stops browsing. The first thing you should do is increase the timeout values in Tools>Preferences>Exploration. Flash ads and banners could also be slowing down the process. For this you should disable everything you do not need for the extraction: Right-click on the page and choose Options. Disable at least images and plugins, even Javascript, if you can (don't forget to reactivate afterwards).

Auto-Explore Functions and Fast Scraping are slower in the current version than in the previous. Why is that?
They are not, in fact. The program's exploration functions work exactly the same way. It is possible, though, that your preference settings have changed during the upgrade. Temporization and pause settings should actually be more precise and reliable than in previous versions. You can fine-tune all this in Tools>Preferences>Time Settings. Another recent preference which may have an impact on the exploration speed is 'Bypass Browser Cache' in the 'Advanced' panel: not using the cache does slow the browsing down, so you may want to set it to 'Never'. If, after this, you are still experiencing performance issues, consider disabling processes you may not need by right-clicking on 'page' in the left side bar.

The next page button functions correctly but when trying to do a Browse to capture the information, the application runs only 2 pages then stops. Why?
- Cause: the next page link is probably a javascript link and it is probably the same in all pages, so the program thinks this URL has already been visited and stops the exploration.
- Solution: there is a preference (Tools>Preferences>Exploration) just for this. Uncheck "Only visit pages once...". **Important:** *Do not forget to check it back afterwards* or your next Dig would probably last forever and bring back huge amounts of redundant data.

I have made a scraper which works fine on the page I want to scrape, but when I do a browse and set the 'scraped' view to collect the data, it grabs the data of the first page over and over again. What is happening?
You are probably trying to scrape information from AJAX pages where the data is dynamically added to the page by Javascript scripts. You need to set the type of source to be used by your *scraper* to *Dynamic*. When you do, the source code of the page will be displayed on a pale yellow background. *Note that you will probably have to adapt your scraper if it was created for the Original source, as the code may have changed slightly.*

When I browse through thousands of urls from a directory of queries, the program sometimes slows down dramatically or even freezes. How can I avoid that?
First, check that your machine resources are not too limited (particularly in terms of RAM: 4GB can be small for very large explorations, 8GB is more comfortable and with 16GB, you should never experience RAM problems). Close the error console if it is open as it consumes RAM and slows the process. Check that the datasheet or the Catch in which the extracted data goes is not sorted because, above a few thousand rows, it becomes pretty heavy to sort all incoming lines of data. Also check that you do not have too many applications working at the same time.
Then, a good precaution is to ask the application to move the results to the Catch (choose 'Auto-Catch' in the views bottom panels --*ALWAYS USE AUTO-EMPTY IF AUTO-CATCH IS ON, or you will end up with millions of useless lines in your catch - this is also true in a macro: if you send data to the catch or to an export file during the execution, make sure that data is deleted from the datasheet or it will accumulate and be re-sent every time, increasing the size of the destination exponentially and slowing down the process progressively until it stalls*). The Catch is saved regularly and will not disappear in case of problem.
This being said, if the application freezes for more than say a mintue or even crashes, there is a reason. It should not happen. So if you have problem cases that can be reproduced, please tell us and we will do our best to correct them.
In any case, for large and broad explorations, it is always a good idea to disable everything you do not need for the extraction: Right-click on the page and choose Options. Disable at least images and plugins, even Javascript, if you can (don't forget to reactivate them afterwards). It will save a lot of processing time and ignore many potential causes for problems. What causes problem could be badly written flash ads or looping scripts, in particular, trying to update content or banners in real time.

During an exploration or an extraction process, I sometimes see an alert saying that a script is not responding. What do I do?
Some extraction processes can be demanding and their execution can be long on very large Web pages or textual files that you load. You have several options at this point: Click on the Cancel button if you believe that this is a one-time problem on an uninteresting oversized page, click Continue (several times, if necessary) if you believe that there is data to be found but this is an exceptional case, Cancel the process and increase the time allowed for script execution in the General preference panel if this is happening frequently. *Note that it is better to have a value (long if you wish) rather than setting it to unlimited: setting the preference to 0 (or checking the don't-show box in the alert) means that you will not be prompted for this in the future. It can be a problem in case of a real bug on a page: you would have to eventually force-quit the application.*

Importing Data

How can I import lists of links (URLs) or other strings into OutWit Hub?
There are many different ways to do this. Here are a few:

- Create a new directory in the ***queries*** view, copy the links from whatever application they are in, click on the right part of the window in the queries view and paste the links.
- Put the links into a text file (.txt) separated with returns and ***open*** the file from the File menu. If the file only contains URLs, the program will ask you if you want to create a directory of queries with them.
- If you want to explore a limited number of links (up to a few hundreds), copy the links from whatever application they are in (you can also copy HTML source code or simple text containing URLs), right-click in the ***page*** view of the Hub and choose Edit>Paste Links.
- Put the links, HTML code or any text that contains URLs into a text file (.txt or .csv), and ***open*** the file from the File menu. (Note that on some systems, the program may try to open .csv files with another application. In this case, just rename your file with the .txt extension.) If you haven't asked to send the URLs to a directory of queries, you will find your them in the ***links*** view and the text in the ***text*** view.
- Drag them directly from another application to the ***page*** or ***queries*** view of the Hub,
- If they are in a local HTML file, simply open the file from the File menu and you will be able to process it with the Hub as any Web page.

When links are in the Hub's links view, you simply need to select them, right-click on one of them and select 'Send to Queries' to create a directory of URLs.
Once your links are in a directory of queries, you can use them any way you like: in a macro, for instance, or doing an automatic exploration directly from the right-click menu (Select those you want, right-click on one of them and choose Auto-Explore Pages>Browse..., Auto-Explore Pages>Fast Scrape..., Auto-Explore Pages>Fast Search for Contacts, etc.).

How can I import CSV or other tabulated data into OutWit Hub?
Simply open the file (.txt, .csv ...) from the File menu. (Note that on some systems, the program may try to open .csv files with another application. In this case, just rename your file with the .txt extension.) If the original data was correctly tabulated, you should find the data well structured in the ***guess*** view. If the data was less structured, well, the Hub will do what it can.

How can I convert a list of values into a String Generation Pattern?
If the values are in one of the Hub's datasheets, just select them, right-click on one of them and select "Insert Rows...". If they are in a file on your hard disk, simply import them into a directory of queries (see above) and do the same.

Extracting Data

I see the information I want on the page but it is not in the source code... How is this possible?
You are probably looking at the original source code of a dynamic AJAX page. The information is added to the page after the page is loaded. For this type of page, you need to work with the 'Dynamic' source code. Set the type of source to be used by your *scraper* to *Dynamic*. When you do, the source code of the page will be displayed on a pale yellow background. *Note that the Light edition of OutWit Hub cannot scrape dynamic data.*

I would like to extract the details of all the products/events/companies in this site/ directory/list of subsidiaries... Could you please advise me on how to do that?
Unfortunately this is the purpose of the hundreds of features covered in the present Help, so it is difficult to answer in one sentence, but the general principle is this:
Go through the standard extractors (documents, lists, tables, guess...) by clicking in the left side panel. Either you find that one of them gives you the results you want, --in which case it is just a matter of exporting the data-- or you need to create a scraper for that site. In the second case, you first need to go to one of the detail pages, build a scraper in the 'scrapers' view for that page, test it on a few other pages. Then go to the list of results you need to grab and have OutWit browse through all the links and apply your new scraper. This can be done in two ways: either by actually going to each page ('browse' or 'dig' or a combination of both if you have a Pro, Expert or Enterprise editions) or by 'Fast Scraping' them (applying your scraper to selected URLs --right-click: Auto-Explore>Fast Scrape in any datasheet-- or 'Fast Scrape' in a macro).

The program doesn't find all the email addresses in this Website, Why is that?
There are several ways to have OutWit look for emails in a site. The fastest is to select Fast-Search For emails>In Current Domain, either from the Navigation menu or from the popup menu you get when you right-click on the page. This method, however, doesn't explore all pages in the site. It only looks for the most obvious (contacts, team, about us...) pages that can be found. If you want to systematically explore all pages in a site, you will have to use the Dig function, within domain, at the depth level you wish.

Why doesn't the program find contact information (phone, address...) for some of the email addresses?
First, of course, the info has to be present in the page. Then, if it is there, no technology allows for perfect semantic recognition. An address or a phone number can take so many different forms, depending on the country, on the way it is presented or on how words are abbreviated, that we can never expect to reach a 100% success rate.
Email address recognition is nearly exhaustive in OutWit; phone numbers are recognized rather well in general; physical addresses are more of a challenge: they are better recognized for US, Canada, Australia and European countries than for the rest of the world. The program recognizes names in many cases. As for other fields like the title, for instance, automatic recognition in unstructured data is too complex at this point and results would not be reliable enough for us to include them unless they are clearly labeled. We are constantly improving our algorithms so you should make sure to keep your application up-to-date.
In the meantime, if automatic recognition is not sufficient, the way to grab precisely the data you want in the format you want, is to create a custom scraper.

I am observing the progress and I see that no new line is added for some pages when I am sure there is an email address or other info that should be found. Why is that?
This page (or one containing similar info) was probably visited before. Results are automatically deduplicated. This means that if an email address --or just a phone number or physical address-- has already been found, the row containing this data will be updated (and no new row, created) when a new occurrence is found.

Exporting Data

What is the maximum number of rows of data OutWit Hub can extract and export? After a certain number of rows, when exporting, I get a dialog telling me a script is unresponsive. What should I do?
In our tests, we have extracted and successfully exported up to 1.3 million rows (of two or three columns). Obviously, the limit varies a lot from system to system, depending on the platform, the RAM, the number of columns, the amount of data in each cell, etc. (Avoid very large numbers of columns: this is something OutWit particularly dislikes. 20 or 50 columns is fine, 100 is usually OK, more than 150 or 200 can make performances collapse dramatically.) When exporting more than 50,000 or 100,000 rows, you may see unresponsive script dialogs, even several times in a row, when you click on Continue. There is a checkbox to stop this dialog from coming back.
(Note that Excel XML export is always much more demanding than CSV or TXT. RAM problems are more likely to happen with Excel than with CSV or TXT as, for the last two, export files are split when they exceed 250MB. You can lower this limit by typing about:config in the address bar and modifying the preference called 'extensions.outwit.export.maxFileSize'. Important: use caution when you change preferences this way. They will alter the behavior of the application and they will not be reverted to factory settings if you use the 'Restore Original Preferences' button in the preference panel.)
A recently added directive, available in the Expert and Enterprise editions, allows to break down the export load into several files: #exportAndDeleteEvery# allow you to export as soon as the chosen number of rows is reached. Releaving the application memory for very large extractions. This may produce large numbers of files in case of extensive scraping, so the ultimate step is to use the Enterprise edition and, using this directive, send the data directly to an SQLite database.
Finally, don't forget that you can simply move your results to the catch and save the catch itself in a file if you need to reuse the contents or just for backup purposes (File Menu). A catch file can only be read again in OutWit Hub but saving is always much faster than exporting the data.

User Interface

How do I make a hidden column visible in a datasheet?
In the top right corner of every datasheet in the application is a little icon figuring a table with its

header: the Column Picker. If you click on this icon, a popup menu will allow you to hide or show the different columns of the datasheet. Only visible columns are moved to the Catch and exported by default (this behavior can be changed with a custom export layout).

What is the Ordinal ID?
The Ordinal column is hidden by default in all datasheets. Use the column picker (icon at the top right corner of any datasheet) to display it. The Ordinal ID is an index composed of three groups of digits separated by dots. In browse mode, the first number is the number of the page from which the data line was extracted (it can only be higher than 1 if the 'empty' checkbox is unchecked.). The second number is the position of the data block in the page (can only be more than 1 in 'tables', 'lists', 'scraped' and 'news' views). The last number is the position of the data line in the block (or in the page, if there is only one data block in the page). In fast scrape mode, the first digit is the scrape execution, and the second is the page number. (If multiple queries where sent from a directory of queries, this second digit matches the order in which the queries were sent, not always the order in which responses were received).

Install

I just purchased the product. Where do I enter my serial number?
If you haven't already downloaded and installed the free version, do so from the home page of outwit.com, then run the application. In the menu bar at the top of the screen, the rightmost menu is 'Upgrade' if no serial number has been entered yet and 'Registration' if a key has already been entered. In this menu, choose 'Enter Serial Number' and enter the key. (To avoid errors, do not retype email and serial number. Instead, copy and paste them from the email you received from OutWit when you ordered.)

I do not manage to enter my serial number in the Registration Dialog of OutWit Hub. The program keeps saying the key is invalid.
Your key was sent to you by email when you purchased the application. It is a series of letters and digits similar to this: 6YT3X-IU6TR-9V45E-AFS43-89U64. It must not be confused with the login password to your account on outwit.com which was also sent to you by email (if you miss one of these email messages, please check your spam box).
If you are wondering whether the Hub you are using is the Light, Pro or Expert & Enterprise editions, you will simply find the answer in the window title. Up to now, we haven't had a single case where a valid serial number would not work. You might be experiencing a very rare bug but this seems very unlikely after several years. The most likely causes are the simple ones: First make sure you have the right program and that you are not trying to enter a Sourcer Pro key in OutWit Hub, for instance. The key needs to be entered exactly like it is in the mail you received. So, either you are not typing it precisely right (in which case you should simply copy and paste the email address and the key from our original mail) or you are typing something completely different (the login to your outwit.com account, for instance?). If you have changed email addresses since you purchased your license, remember that the one to use is the one with which you originally placed your order.

Troubleshooting

This section gives you important info on the way to find and backup your profile files and to revert to factory settings. You should also read Help>Help>Standalone Application for more info on the same topic.

All my scrapers have disappeared. Help!
Don't panic! If, for some reason, your profile directory (or folder on mac) was not found when the program started or if it doesn't have the name that was expected, OutWit Hub will create a new install with a blank profile. However, unless you have explicitely deleted the previous directory, your old profile and all your automators are not lost: they are sitting in your old profile directory, in a file called User_Gear.owc. You can either learn below how to work with multiple profiles or simply open the old User_Gear.owc file from the application with File>Manage Automators>Import Automators From... Read the following and the help page on the Standalone Application to know how to locate and manage profiles.

On OutWit Hub For Firefox, I have been experiencing new issues recently: unresponsive scripts, timeouts, strange behaviors on pages that used to work fine... what can I do to revert to factory settings?
We are not aware of incompatibilities with other add-ons but it can always happen, some of your

Frefox preferences could also have been changed by another extension or files may have been corrupted in your profile. You can try to create a blank profile and reinstall OutWit Hub (or other OutWit extensions) from outwit.com. This will bring you back to the initial state. Here is how to proceed on Windows:
http://kb.mozillazine.org/Creating_a_new_Firefox_profile_on_Windows
and on other platforms:
http://support.mozilla.com/kb/Managing+profiles

Can I create a new profile in OutWit Hub Standalone?
With the standalone version, the principle is almost exactly identical to the way it works in Firefox (see above paragraph).
Windows: click "Start" > "Run", and type :
"C:\Program Files (x86)\OutWit\OutWit Hub\outwit-hub.exe" -no-remote -ProfileManager
Macintosh: Run the Terminal application and type :
/Applications/OutWit\ Hub.app/Contents/MacOS/outwit-hub -no-remote -ProfileManager
Linux: open a terminal and type :
[path to directory]/outwit-hub -no-remote -ProfileManager

If you need instructions to go further, refer to the profile manager instructions for Firefox:
http://support.mozilla.org/en-US/kb/profile-manager-create-and-remove-firefox-profiles

Where is my profile directory?
In OutWit Hub (Standalone or Firefox Add-on), if you type about:support in the address bar, you will get a page with important information about your system and configuration. In this page, you will find a button that will lead you to your current profile directory with a name like "u3p9be0z.Default". If you have multiple profiles or if you are looking for an old profile, look in the parent directory called "Profiles". Among the files you will see in the profile directory, the ones with .owc extensions are Catch files, and files ending with .owg are User Gear files (the User Gear is the database where all your automators are stored). You can back these files up or rename them if you plan to alter your profile.

...

Frequently Asked Questions (Advanced) - WORK IN PROGRESS

Here are additional answers to questions which require advanced commands. *[updated:6/8/15]*

Address Bar Commands

I have read about commands that can only be typed in the address bar. Which are they?

- about:config
- about:support
- about:memory
- outwit:restart
- outwit:debug
- outwit:registration

Troubleshooting

What is New in OutWit Hub v8

Here are the main additions and changes between v7.x and v8.x of OutWit Hub.
(The minimum required license type is indicated between parantheses.)
This page is just an overview, please look for usage detail in the corresponding Help sections. *[updated:6/11/19]*

- Scrapers
- Data Extraction & Enhancement
- Exporting Data
- Automatic Exploration
- Editors
- User Interface

Scrapers

New Directives and Advanced Replacement Functions

Here are some scraper directives and functions that were added in version 8.0:
(check Scraper Editor Help for details and edition) :

- #emptyDirectory# Empties the first directory of the queries view matching the passed name.
- #splitField# Splits the passed field as a post-process, using the values in the separator and labels columns. (Can allow consecutive splits.)
- #decodeEntities# Decodes HTML entities (like &amp; or &gt;) to their plain text equivalent.
- #decodeURL# Decodes URL encoded characters (like %20) to their plain text equivalent.
- #save# Saves the string extracted by the scraper line to a separate text file.
- #screenshot# Saves a screenshot of the current page into a file using the passed file name.
- #hideNodes# Makes the nodes matching the passed css selector invisible.
- #scrollBy# Scrolls the page loaded in the OutWit Hub browser by the passed number of pixels.
- #resetPrefOnStop# Reset the passed preference to its default value at the end of the scrape process.
- #uniqueField# Makes sure that no duplicate values are extracted for the specified field(s) during the same exploration. (An alternative to deduplication while scraping, in case volumes are too large to post-process it.)
- #setValue# Sets the value of the <select> or <input> HTML block matching the format column, to the value passed in the replace column.
- #restartEvery# Sets 'auto-explore on startup' flag to true and restarts the application, every n pages or seconds.
- #uncheckURLInQuery# Unchecks the 'OK' checkbox of the first line containing the current URL in the passed query directory.
- #uncheckItemInQuery# Unchecks the 'OK' checkbox of the first line containing the string extracted by the scraper line in the passed query directory.
- It is now possible to set the field name with a variable in the #default# directive.
- #readFromQueries# Reads the next active string from the passed query directory and stores its value in the passed variable, then unchecks the line in the query directory.

- #switchTo# Changes the current view to the value set in the replace column.
- #reapply# now accepts parameters for the number of applications and the delay between them.

- #adler32()# Used in the replacement column, allows you to generate a short hash from the string extracted by the scraper line. (This can be useful for deduplication although it is not 100% reliable as, even if it is unlikely, two different strings can result in the same hash.)
- #encodeBase64()#, #decodeBase64()# Converts the string extracted by the scraper line into a base64 encoded string or decodes it into plain text.
- #decode()# Decodes the string extracted by the scraper line into plain text, trying several algorithms.
- #unique()# Only returns the string extracted by the scraper line if the values is unique during the same exploration. (An alternative to deduplication while scraping, in case volumes are too large to post-process it.)
- #WEEK# was added to the time variables. Returns the week number in the year.
- #LAST-POST-QUERY# returns the last POST query send. #LAST-POST-QUERY#param# returns the value of the passed parameter in the last POST query sent.

Data Extraction & Enhancement

Faster volume scraping
Faster start preparation and end of process cleaning in large volume Fast Scrapes.

Periodic Scraper Executions (Expert & Enterprise)
When scraping a self-updating AJAX page, the #reapply# directive now allows to do the extraction n times at the frequency you choose.

Contact Recognition (Pro & Above)
The contact recognition module and its dictionary were enhanced, lax recognition and dummy email addresses elimination, improved.

Recognition (Expert & Enterprise)
Improved dictionary of multilingual words, acronyms and roots, frequently used in company names addresses etc. to enhance recognition.

Data Refining: Editors & Datasheet Functions

Several tools were added to the right-click menu on datasheets:
Insert Index Column: inserts a column with an incremeted index.
Duplicate Column: copies values from a column to another.
Indexed Duplicate Column: inserts a new column with values from another where duplicate cells and suffixed with an index.
Copy from Column...: copies values from a column to another.
Select if in...: selects rows in a datasheet if they are found in another datasheet. Particularly interesting to identify the rows in an extraction that have already been extracted and sent to the catch.
Duplicate cells are now underlined in the scraper editor to be clearly identifiable.
Copy & paste allows to transfer content between scrapers or jobs.

SQLite Data Import (Enterprise)
It is now possible to open a SQLite database file into OutWit Hub. This can be very useful when resuming previous extractions, to merge several files or simply to refine the data after extraction.

Exporting Data

Added the semicolon-separated CSV export format for countries that use commas as decimal separators.
Multiple fixes and enhancements in export functions.

Automatic Exploration

New URL Edition Tools

In addition to the URL Editor and the String Generation Panel (Right-Click>Insert>Insert Rows...), a Search Query Builder is now available in the Tools Menu and as a Toolbar button. It allows to generate complex queries for the most used search engines.

Enhancements and Fixes in POST Query Generation Syntax

#HEADER# allows to add custom parameters to the header of the query (see POST query format). #CHARSET# defines the encoding, #TYPE#, the contentType and #REFERER#, the referrer.

User Interface

Application Info Box

Added an info box at the bottom left corner of the application window.

Display Mode Buttons

A line of display mode buttons located at the bottom of the window, left of the status bar, allow you to easily toggle on and off the display of images or videos, the highlighting of nodes or series of links, the activation of plugins and javascript.

Error console

The error console is automatically closed if it seems to slow down the process of a fast scrape.

Automator Property Dialog

Multiple selection is now possible when opening the automator property dialog, allowing to edit a field for multiple items at once.

And more...

v8 includes many more enhancements and fixes that are not listed here, improving overall security, reliability and stability of the program.

New in OutWit Hub v7

Here are the main additions and changes between v6.x and v7.x of OutWit Hub.
(The minimum required license type is indicated between parantheses.)
This page is just an overview, please look for usage detail in the corresponding Help sections. *[updated:1/30/18]*

- Scrapers
- Data Extraction & Enhancement
- Exporting Data
- New OutWit Fetcher Edition

Scrapers

Over 60 New Directives

A very long list of scraper directives and functions was added in version 7.0:
We cannot list all of them but here is a selection (check Scraper Editor Help for details and edition) :

- #exportEvery#n#, #exportAndDeleteEvery#n#, #catchEvery#n#, #catchOnStop# Catches or exports the extracted data when you desire during the process.
- #abortAfter#, #abortAfterNPages#n#, #abortAfterNResults#n# Aborts the current extraction after a text is found or a certain number of pages or results have been reached.
- #decodeJSCharcodes#, #zapGremlins# to decode hexadecimal, remove unwanted control or invisible characters, correct badly encoded characters, etc.
- #clearForms#, #clearAllHistory#, #clearBrowsingHistory#, #clearCookie#, #clearCookieEvery#n#, #clearCookieIf#, #clearCookiesEvery#n#, #clearCookiesIf#, #clearCookiesIfNot# allow you to manage history and cookies from within the scraper.
- Use #autoEmpty#, #autoCatch#, #emptyOnDemand#, #deduplicate# to set the value of the scraped view options from a scraper.
- #keepForms#, #removeScripts#, #removeTags#, #allFrames#, #originalHTML#... allow you to determine exactly how you want the source before the scraper is applied.
- #replaceInField#fieldName# replaces value (litteral of RegExp) in a given field at the end of the process.
- #fieldGroup# Makes sure that the fields indexes in a same group are incremented together even if some of the fields are empty.
- #oneRow# Makes sure that all extracted data in the page will be presented as a single row in the datasheet.
- #allowCrossDomain# removes js restrictions, which is sometimes useful to simulate clicks and other interactions with the page.
- #rename#, #unzip# give you post-processing access to files that you have downloaded.
- ... and many, many more.

New advanced replacement functions (Expert & Enterprise)

The #match()# function, used in the replacement column allows you to search the other occurrences of a string (or matches of a RegExp) that you grab (or build) from the page itself. It allows very powerful conditional extractions.

Limiting multiple & duplicate results (Expert & Enterprise)

The syntax **myFieldName<n, myFieldName>n, myFieldName=** in the description column allows you to manage multiple results, duplicates etc..

Data Extraction & Enhancement

News / RSS feed Extraction (Pro & Above)

Improved recognition and extraction of rss feeds, publication dates in more locales, addition of universal identifier (guid)...

Contact Recognition (Pro & Above)

The contact recognition module was further enhanced, lax recognition and dummy email addresses elimination, improved.

Name Recognition (Expert & Enterprise)

A large dictionary of multilingual words, acronyms and roots, frequently used in company names addresses etc. was added to enhance recognition.

Exporting Data

Extract millions of rows (Enterprise)
Used in the Enterprise edition, the #exportAndDeleteEvery#n# scraping directive can define an SQLite database as the destination (using a filename with the .sqlite extension), allowing you to process and store extremely large volumes of data.

Saving Preferences (Expert & Enterprise)
You can now save (and restore) the state of preferences in (from) a directory of the queries view.

New OutWit Fetcher

If you need copies of OutWit Hub Expert for a fraction of the price, just to run extractions: OutWit Fetcher.
OutWit Hub still comes in three different editions (license levels): Pro, Expert and Enterprise but we now propose a streamlined version of the Hub that can do Web explorations and run scrapers but has no editing capacities. Don't hesitate to enquire about this on the customer support system.

And much more

v7 brings many additional enhancements and fixes that are not listed here, improving overall security, reliability and stability of the program.

New in OutWit Hub v6.x

Automators

Projects (Pro & Above)
Pro users can now organize their automators (scrapers, macros, jobs, queries), grouping them by projects and saving them as coherent collections.

New Directives (Pro & Above)
A large series of scraper directives and functions was added to the pro version:
Use #autoEmpty#, #autoCatch#, #emptyOnDemand#, #deduplicate# to set the value of the scraped view options from within a scraper.
The #default# directive allows you to set a default value to all fields, #default#fieldName# sets a default value for the passed field name.
Use #pauseBefore# to instruct the program to wait for the passed number of seconds before extracting the data...
#checkIfURL# and #checkIfNotURL# directives allow you to include URL-based conditions in a scraper.
#SECOND# to #FIFTH# were added to the replacement functions in scrapers, allowing to extract the corresponding occurrence of a matching string.
The #LOCALIP# replacement function allows you to access the current local IP from scrapers (can be useful when rotating proxy IPs)

New Directives (Expert & Enterprise)
The #storeVariables# directive makes variables set in a scrape available to subsequent scrapes.
#scope# defines the scope of the extraction in fast scraping/diging mode

with the Expert edition (outside or within domain, all links or with a depth of 1 or 2).
#coalesceOnStop# instructs the program to merge extracted data rows, grouping them by the passed field.
#deduplicateOnStop#criterionColumnName# Does a smart deduplication of the extracted data (row by row) in the datasheet, once the current automatic exploration is achieved. (This prevents the deduplication to slow down the whole process.)
#deduplicateWithinPage# does a smart deduplication of the extracted data (row by row) for each scraped page, before sending the results to the datasheet. (This prevents the deduplication of potentially tens of thousands of rows to slow down the whole process.)
#scrollToEnd#cssSelector# was added, in order to address a specific HTML element and scroll down within this element. Very useful for recent AJAX interfaces.
The #encodeURL()# replacement function was added to encode special characters as they should be in a URL (the space character becoming %20, etc.). This is required in some cases when adding URLs to the queue of pages to explore.
#base64()# allows to convert a small image into a self-contained data element using the data: URI scheme.

Data Extraction & Enhancement

Email Address Recognition (Pro & Above)
The email recognition module was enhanced. It now allows for diacritic characters, more dummy email addresses (user@example.com...) are eliminated, lax recognition (jackie at mysite dot com...) is much more efficient.

Names and Genders (Expert & Enterprise)
A preference instructs the program to create an additional Gender column when using the Insert First/Last Name function in the right-click menu. This column will contain the string defined in the preference (i.e. "Dear Mr", "Ms", "Herr", "Chère Madame"...) when the gender is recognized and a fallback value ("Dear Customer"...) otherwise.

Name/Gender replacement functions (Expert & Enterprise)
New directives allow to get First Name, Last Name, First & Last Names, Gender from an extracted string directly in the scraper.

Word Count (Expert & Enterprise)
The words view now includes a text box where you can type or paste the words to count in the page. You can also paste a whole text to count common words between the Web page and this reference text.

Exporting Data

General enhancement and optimization of the Export module (All)
The export module was refactored and optimized in v6.0, fixing bugs, enhancing data cleaning and performance and adding features like additional preference settings for SQL exports (list of fields in INSERT statements, VARCHAR(xxx)...)

Appending extractions to previously exported data (Expert & Enterprise)
Macros can append extracted data to an existing txt, csv or SQL file.

FTP upload (Expert & Enterprise)
Extracted data can be uploaded to an FTP server, adding an index to the file name, if the same name already exist, or overwriting it. When FTP upload is selected as a destination in a macro, the FTP server info set in the advanced preferences is proposed by default.

SQLite Export (Enterprise)
OutWit Hub Datasheets and Catch can now be exported directly to an SQLite database. (SQLite is by far the most widely deployed SQL database engine in the world, installed on billions of computers, smart phones, tablets, TVs etc. It is powerful, easy to set up and use and it can of course be exported to any other database format.)

Automatic Exploration

Shuffle Rows (Pro & Above)
Added a Shuffle function to the right-click datasheet menu which allows, (in particular in the 'queries' view) to randomly reorder rows to avoid sending queries to a server in numerical or alphabetical order.

Check HTTP headers before querying a page (Pro & Above)
Added a prference to instruct the program to check the page header before loading it, in order to avoid errors and login dialogs that could block an automatic exploration.

Stacks in Database (Enterprise)
Added "Stacks in Database" preference which instructs the application to store the current exploration stacks (urls to visit, already visited urls...) into a database instead of in the RAM. This configuration is very interesting for large volume explorations and extractions that span over several days. It can multiply the maximum number of processed web pages you can process by a factor of 5 to 10 and fast scrapes of hundreds of thousands to several millions of pages become possible without running into memory limitations.
A series of commands was added to directly alter the exploration stacks: backToQueue, addToVisited, removeFromVisited, addToURLsToVisit, removeFromURLsToVisit, emptyStack, which can be accessed both from the datasheet right-click menu or typed in the address bar with the prefix outwit:

Main features in the different editions of OutWit Hub

Here is a list of the program's main navigation an extraction functions and in which editions they can be found.

- License levels overview
- Data Extraction
- Automatic Exploration Tools
- Automators
- Exporting Data
- Advanced Automation

License levels & setup

From basic scraping of a Webpage to complete automated workflows on multiple machines

	Light	Pro	Expert	Enterpri
Overview General description of the license	Free, limited but operational for small data extraction jobs.	Recognized, solid, inexpensive yet powerful scraping tool, with a wide spectrum of exploration and extraction features.	Power-user scraping tool with all advanced functions for fast, high-volume scraping and complex cases.	Full power ver large proje multiple-insta multi-user v OutWit's top packag
Concurrent Users/ Uses Maximum number of users or instances of the program running at the same time	1	1	1	3+ (see vo licensing
Installs Number of software installs per license at one point in time	Any platform, Mac, Windows, Linux	3 machines, any platform	3 machines, any platform	9 machines platforr
Setup Type of install	Standalone application to simply download & install on laptop, desktop or serve			
Team deployment Sharing of automators	no	XML exports	XML, native DB format exports	LAN/WAN cen automator da XML, nativ format exp

Data Extraction

Base Extractors
The panel on the left side of the screen contains a series of automatic extractors

	Light	Pro	Expert	Enterpri
Links Automatic link extraction	limited to 100 results	unlimited extractions	unlimited extractions	unlimited extr
Documents Document extraction & download	no	unlimited extractions	unlimited extractions	unlimited extr

Images Image extraction & download	limited to 100 results	unlimited extractions	unlimited extractions	unlimited extr
Contacts Extraction of contact information	limited to 10 results	limited to 5,000 results	limited to 100,000 results	unlimited extr
Lists Extraction of HTML lists	limited to 100 data rows	unlimited extractions	unlimited extractions	unlimited extr
Tables Extraction of HTML tables	limited to 100 data rows	unlimited extractions	unlimited extractions	unlimited extr
Scraped Custom data extractions using scraper	limited to 100 data rows, simple scraper	unlimited extractions, remplace/split functions, base set of functions & directives	unlimited extractions, remplacements/ splits, advanced set of functions & directives	unlimited extr remplacem splits, all functio directive shared scr databas
Text Simple text extraction	yes	yes	yes	yes
Words Recurring words & groups of words	no	unlimited extractions	unlimited extractions specific word count	unlimited extr specific worc
Source Colorized source	Original Source Only	Original, Dynamic	Original, Dynamic, All Frames	Original, Dyna Frames
News RSS feed extraction	no	unlimited extractions	unlimited extractions	unlimited extr

Automatic Exploration Tools

Automation Functions
The toolbar and navigation menu contain a number of automatic exploration tools

	Light	Pro	Expert	Enterpri
Browse Automatic browsing through series of result pages	yes	yes	yes	yes
Dig Exploration of pages and the links they contain at a desired depth level	yes	yes	yes	yes
Advanced Dig functions Choice of scope and depth exploring URLs	no	yes	yes	yes

Fast Dig & Browse Fast mode exploration in macros	no	no	yes	yes
Fast scrape Fast scraping without rendering pages	no	yes: on list of URLs	yes: list of URLs, whole domains and self-navigating	yes: list of whole domai self-naviga
Fast search for contacts Fast-search for contacts throughout domains	no	yes: on list of URLs	yes: list of URLs, whole domains and self-navigating	yes: list of whole domai self-naviga
Check HTTP headers (pref.) Checks headers before loading pages to avoid blocking dialogs in explorations	yes	yes	yes	yes
Stacks in Database (pref.) Stores exploration stacks (urls to visit, visited...) in database instead of RAM.	no	no	no	yes
Interrupt/Resume Interrupt and resume exploration when stacks in DB	no	no	no	yes

Automators
The panel on the left side of the screen contains a series of automatic extractors

	Light	Pro	Expert	Enterpri
Projects Grouping automators by projects	no	yes	yes	yes
Queries Directories of links & queries	no	yes	yes: can be automatically populated from a scraper	yes: can automatic populated f scrape
Scrapers Advanced scraper functions	simple extraction between markers, no directives or functions	basic set of extraction and data processing functions	advanced set of extraction and data processing functions	all extractio data proce functions, p sharing
Macros Exploration/ extraction process automation	no	yes	yes: with the possibility to feed/ augment the list of URLs to explore during the exploration	yes: with possibility to augment the URLs to ex during t explorati
Jobs Periodical execution of macro(s)	no	yes	yes: with the possibility to inlude several macros,	yes: with possibility to several ma

			system commands and functions in a job	system comm and functions

Advanced Automation

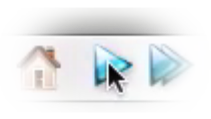
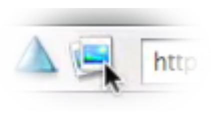
	Light	Pro	Expert	Enterpri
AJAX Dynamic source scraping	no	yes	yes: extracts data from multi-frame, AJAX-modified source	yes: extract from multi-f AJAX-modifiec
POST requests Simple POST request generation format	no	yes	yes: can be automated within a scraper	yes: can automated w scrape
Query interception GET, POST query interception & alteration	no	no	yes	yes
Generation Patterns URLs, string generation patterns	no	yes	yes: can be automated within a scraper	yes: can automated w scrape

Exporting Data

Extracted data can be previewed and exported to the following formats using the Export panels

	Light	Pro	Expert	Enterpri
HTML HTML Table. Can be opened in a navigator, can include images	yes: limited	yes	yes	yes
HTML Detail HTML List, by paragraphs. Can be opened in a navigator, can include images	yes: limited	yes	yes	yes
Excel Export file can be directly opened in Microsoft Excel	yes: limited	yes	yes	yes
CSV Can be imported in most programs, including Excel and database systems	yes: limited	yes	yes: can append exported data to existing file	yes: can ap exported da existing
TXT Can be imported in most programs, including Excel and database systems	yes: limited	yes	yes: can append exported data to existing file	yes: can ap exported da existing

XML Starndard XML	yes: limited	yes	yes: can append exported data to existing file	yes: can ap exported da existing
vCard Used for contacts. Can be opened in most phone, address book systems	yes: limited	yes	yes	yes
JSON JSON object format, usually used for programming purposes	yes: limited	yes	yes	yes
SQL Records are exported as SQL INSERT statements	yes: limited	yes	yes	yes
SQL UPDATE Records are exported as SQL UPDATE statements	yes: limited	yes	yes	yes
SQLite Most used SQL-Based database format	no	no	no	yes
FTP upload Uploading scraped data to an FTP server	no	no	yes	yes

	**Next in series**: Loads the next page in a series *Active when OutWit finds a navigation link to the following page (i.e. if the current page is part of a series, like a result page for a query in a search engine).*
	Browse: Auto-browses through the pages of a series. *Active when OutWit finds a navigation link to the following page (i.e. if the current page is part of a series, like a result page for a query in a search engine). Right-clicking or holding down the Browse button opens a menu allowing you to limit the number of pages to explore. Escape or a second click of the button will stop the browse.*
	Dig: Automaticallly explores the links of the current page. *Active when OutWit finds links in the current page. Right-clicking or holding down the Dig button opens a menu allowing you to limit the exploration within or outside the current domain and to set the depth of the dig. Depth = 0 will browse through all the links of the page, Depth = 1 will also explore the all the links of pages visited. Escape or a second click of the button will stop the dig. (Only links matching the list of extensions set in the advanced preference panel are explored. Some link types are systematically filtered out from the exploration: log-out pages, feeds which cannot be opened by the browser, etc.)*
	Up to Site Home: Goes up to the home page of the current site. *Active when the current page is not the home page of a site. Goes up one level towards the top of the current site's hierarchy.*
	**Slideshow**: Displays the images of the page as a slideshow. *Active when OutWit finds images in the current page. The slideshow can be viewed in full screen or in the page widget. If the current page is part of a series, the slideshow will go on as long as a next page is found.*
	Address Bar: for URLs, macros or search queries. *You can type here a URL to load, a query which will be forwarded to the preferred search engine, or a macro to execute.*

The *Standalone Application*

OutWit Hub exists in two guises: a standalone application and a Firefox add-on.

IMPORTANT NOTE: Because of Mozilla's new signature restriction, the add-on versions of OutWit programs cannot be installed in versions higher than 43 for Firefox and 5.5.5 for Tor. You can find old versions of both programs but the best to remain up-to-date is to use our Standalone applications.

Both (standalone and add-on) are basically the same program and are able to fulfill the same functions. There are however a few specificities corresponding to their natures. Here are the ones worth noting:

(If you wish to get to your OutWit files, please first read the Frequently Asked Questions, Troubleshooting section, for info on the Profile files in both the Standalone app. and the Firefox add-on.)

- **The standalone application can be launched from a terminal:**
  - Windows: "C:\Program Files (x86)\OutWit\OutWit Hub\outwit-hub.exe"
  - Mac OS: /Applications/OutWit\ Hub.app/Contents/MacOS/outwit-hub
  - Linux: run outwit-hub from the location where you unpacked the zip file.

- **In the command line, to run the standalone application from a terminal, you can include the following parameters:**
  - **-url "http://..."** *to load an URL after starting. Using the quotes around the URL is safer, in case of special characters, especially on Window.*
  - **-url "outwit://..."** *to run a Macro As URL (generated by another process or copied from the macro editor).*
  - **-macro xxx** *to execute the macro corresponding to the Automator ID (AID) xxx in your profile (see the list of macros in the macro manager).*
  - **-job xxx** *(Expert & Enterprise editions) to execute the job corresponding to the Automator ID (AID) xxx in your profile (see the list of jobs in the job view).*
  - **-quit-after** *to instruct the application to quit after executing the tasks of the command line.*
  - **-p** *to open the profile manager. (You can also specify a profile name to open directly, without displaying the profile manager: -p profileName)*

- **The standalone profile files with your automators and catch are located by default in (replace XXX by your user directory):**
  - Windows: C:\Users\XXX\AppData\Roaming\OutWit\outwit-hub\Profiles\
  - Mac OS: /Users/XXX/Library/Application\ Support/OutWit/outwit-hub/Profiles
  - Linux: outwit-hub/Profiles

    Your profile folder, located in this folder, has a name like "wzd1klmx.Default User". **Back this directory up** before any operation on your profile. If you wish to do a **clean install**, you need to quit the application and delete (or rename) this profile folder or even the folder two levels above (C:\Users\XXX\AppData\Roaming\OutWit on Windows or /Users/XXX/Library/Application\ Support/OutWit on Mac), if you do not have several profiles that you want to preserve. This will force the program to recreate a blank profile during the next launch.
    After you have re-installed with a blank profile, if you have backed up your automators, you can import them back, using File>Manage Automators.

    *Note: In OutWit Hub (Standalone or Firefox Add-on), if you type **about:support** in the address bar, you will get a page with important information about your system and configuration. In this page, you will find a button that will lead you to your profile directory. The file named User_Gear.owg contains your scrapers, macros, etc. and catch.owc contains the data you placed in your Catch. Your profile directory also contains backup folders where old versions of these files are stored. (In the Firefox Add-on version, your OutWit profile files are located within the Firefox profile.)*

- **The registration file with your serial number is located in a file called OutWit-presets.xml**
  - Windows: C:\Users\XXX\AppData\Roaming\OutWit\OutWit-presets.xml

    - Mac OS: /Users/XXX/Library/Application\ Support/OutWit/OutWit-presets.xml
    - Linux: OutWit-presets.xml (in the folder where you did the install).

  Deleting the presets file (after quitting the application) will reset your installed program to the light version. You will need to reenter your serial number to restore Pro or Expert features.

- **If your license level allows it (multiple-seat licenses, *Enterprise edition*...), you can run several instances of the standalone version concurrently:**

  Start by creating two or three profiles (it is generally not advisable to run more instances on a single machine because of the intensive use of RAM and CPU), then run them separately.

  (In the profile manager, uncheck the 'Use profile without asking' checkbox.)

    - Windows: click "Start" > "Run", and type: "C:\Program Files (x86)\OutWit\OutWit Hub\outwit-hub.exe" -no-remote -ProfileManager and create as many profiles as you want instances. To run instances with a double-click, edit (or create) a desktop shortcut to OutWit Hub (right-click: properties) and add -no-remote -ProfileManager at the end of the Target field.
    - Mac OS: Run the Terminal application and type: /Applications/OutWit\ Hub.app/Contents/MacOS/outwit-hub -no-remote -ProfileManager and create as many profiles as you want instances. If you want to run the instances from the finder with a double-click, duplicate the OutWit Hub application in your Applications folder in as many copies as you want instances. Run these copies, each with a different profile.
    - Linux: open a terminal and type: [path to directory]/outwit-hub -no-remote -ProfileManager and create as many profiles as you want instances. Run each instance with a different profile.

- **The Firefox Add-on allows the Hub to open new browser windows as new Firefox tabs or windows. The standalone version doesn't have this capacity.**

OutWit Hub's Menus

The menus give access to the main features of the application.

The application Menus located at the top of the screen are

- the File Menu
- the Edit Menu
- the View Menu
- the Navigation Menu
- the Tools Menu
- the Help Menu
- the Registration/Upgrade Menu

A contextual menu, the right-click popup Menu, can be used in all datasheets of the application.

The File Menu

***Gives access to the file saving/loading and data export functions*.**

Available options may vary with the *view* and the license level of your product.

Open...

Opens the File Picker Dialog to select one or several files from the hard disk or a local resource. Some file types can be explored and processed by OutWit to recognize and extract content (html, htm, xhtml, xml, txt, csv, owc...). When the selected file can be processed by OutWit Hub, it will be opened in directly OutWit Hub, otherwise, it will be ignored (or, in the case of OutWit Hub for Firefox, it will be open/processed by Firefox). When several files are selected in the Open Dialog, if some or all of them can be explored by OutWit, they will be successively browsed by the program. If one or several files are OutWit Automators or Catch files, they will be imported after confirmation by the user.

Notes about importing data:

- *You can open .html files of course, but also .txt, .sql, .csv... files of many different types and formats and process them with the Hub. The guess view should do a good job recognizing the fields of tabulated files in most cases --if the file is not too exotic.*
- *Putting a list of URLs in a .txt or .csv file and opening it in the Hub is one of the easiest ways to import links for automatic exploration and processing. They will appear in the links view, from which they can be grabbed, sent to a directory in the queries view...*

Save Page As...

Same command as in any browser: Saves the current page on the hard disk. The attached files and images will be saved in a folder called with the name of the page siffixed with "_files".

Download Selected Files

Downloads and saves to the current destination folder on your hard disk, all documents and images found in the selected rows. *(The same option can be found in the datasheet right-click menu.)*

Download Selected Files in...

Downloads all documents and images found in the selected rows, opening the folder picker to let you decide where you want the files to be saved. *(The same option can be found in the datasheet right-click menu.)*

Load a Catch File...

Opens the File Picker to select a Catch file to open on the hard disk or a local resource.

Save Catch File as...

Save the content of the Catch as an OutWit Catch file (.owc) to the hard disk or a local resource.

Export Catch as...

Exports the content of the Catch to a file on your hard disk, in one of the available formats (Excel, CSV, HTML, SQL).

Export Selection as...

Exports the selected data to a file on your hard disk, in one of the available formats (Excel, CSV, HTML, SQL). *(The same option can be found in the datasheet right-click menu.)*

Empty Catch

Deletes the contents of the Catch panel.

Manage User Gear

Allows you to Export or Import the User Gear database, which contains all your automators. This way, you can easily transfer your scrapers, macros... from one profile to the other or from the addon to the standalone version.

The Edit Menu

***Gives access to the application's text and datasheet editing functions*.**

Available options may vary with the *view* and the license level of your product.

Editing Functions

The standard Cut, Copy, Paste, Duplicate and Delete functions apply to the selection. In a datasheet, they apply to rows.

Insert, delete, edit, copy and empty functions are available for cells. Columns can be inserted or deleted.

Insert Row

Inserts an empty row to the current datasheet, before the selected row.

Insert Rows

The Insert Rows function allows you to generate strings using the *Query Generation Pattern* format. Inserts the generated rows to the current datasheet, after the selected row.

Select All

Selects all rows of the datasheet.

Invert Selection

Deselects all selected rows of the datasheet and selects all rows that were not selected.

Select Similar

Selects all rows of the datasheet with content similar to that of the the selected cell. The default threshold used for determining similarity is 40 (0 selecting only identical values and 100 selecting everything). Use the sub-menu items to increase or decrease the threshold and select more or less rows.

Select Identical

Selects all rows of the datasheet with content identical to that of the the selected cell.

Select Different

Selects all rows of the datasheet with different content from that of the the selected cell.

The View Menu

***Gives access to the application's display options*.**

Available options may vary with the *view* and the license level of your product.

Slideshow

Displays the images of the page as a slideshow.

Full Screen

Displays the page in full screen. In this mode, menus disappear. To exit the full screen mode, use the platform function key or press the escape key.

Show/Hide Catch

Displays or hides the Catch Panel at the bottom of the application interface.

Show/Hide Log

Displays or hides the Log Panel at the top of the application interface.

Show/Hide Info

Displays or hides the Info/Message Bar at the top of the application interface.

Switch View Mode

Rolls through the different display settings for the current view: Data only (spreadsheet display), Export Layout only (HTML, CSV...) or a split view with both.

Increase Text Size

Increases the text size in datasheets, managers editors and logs. Menus and buttons will not be affected.

Default Text Size

Reverts to the default text size in datasheets, managers editors and logs.

Reduce Text Size

Reduces the text size in datasheets, managers editors and logs. Menus and buttons will not be affected.

Highlight Series of Links

When checked, the program will highlight links of the same group or level, to simplify automatic exploration.

Show Exploration Button

When checked, the program will display a button in the page with which you can display the main automatic exploration functions in a simple click. (The exploration menu can also be displayed by right-clicking on the page.)

Windows

Lists and gives access to the windows currently open in Firefox.

Views

Lists and gives access to the Hub's views.

The Navigation Menu

***Gives access to the application's Navigation options*.**

Available options may vary with the *view* and the license of your product.

Fast Search for Contacts and ***Auto-Explore Pages*** are also accessible using the right-click menu on the page or the Exploration Button.

Back

Goes back one page in the navigation history.

Forward

Goes forward one page in the navigation history.

Next in series

Loads the next page in a series *(more info on the next page function.)*
Active when OutWit finds a navigation link to the following page (i.e. if the current page is part of a series, like a result page for a query in a search engine).

Fast Search for Contacts

The program sends queries to the site(s) and searches for emails without loading the pages in the browser. Available options in this sub-menu vary with the current page and context. They include:

In Current Website

The program sends queries to the current site and searches for contacts without actually loading the pages in the browser. Not all pages are explored. OutWit Hub tries to locate the ones that are likely to include contact information.

In All Links

The program sends queries to the URLs found in the current page to search for email addresses and contact information.

In Selected Links

The program sends queries to the selected links, searching for email addresses and contact information.

In Highlighted Links

The program sends queries to the highlighted links, searching for email addresses and contact information. *(Hover over the links to highlight series of links.)*

In Linked Websites

The program browses through the pages of the current series of result pages (if any) and sends queries to the external URLs found (linking outside the current domain) to search for email addresses and contact information.

Auto-Explore Pages

The program actually visits and loads each page of a series or selection. Available options in this sub-menu vary with the current page and context.

Browse Selected Links

Auto-browses through the links that are selected in the current page.

Browse Highlighted Links

Auto-browses through the links that are highlighted in the current page. *(Hover over the links to highlight series of links.)*

Browse Series of Result Pages

Auto-browses through the pages of a series.
Active when OutWit finds a navigation link to a following page (i.e. if the current page is part of a series, like a result page for a query in a search engine). Escape or a second click on the Browse button will stop the auto-browse process. Right-clicking or clicking and holding down the Browse button shows a menu allowing to choose the extent of the automatic browse to perform (2,3,5,10 or all pages).

Dig / Browse & Dig Result Pages

Gives access to the Dig sub-menu: The Dig function is a systematic exploration of all links found in a page, in a whole site or in a series of result pages. In order to not visit hundreds of unwanted pages randomly, you can set a number of limitations. You can visit pages if they are within the same domain as the current page, outside the page domain or you can visit any link found. You can also specify the *Depth* of your exploration: Depth 0 is the list of links found in the page, depth 1 also includes all the links found in each visited page and depth 2 does the same one level below. In the *Advanced Settings* dialog, you can combine all these criteria and even set an additional filter with a string (or a *regular expression*) which must be present in the URL for the program to explore it. *(Only links matching the list of extensions set in the Automatic Exploration preference panel are explored in a Dig. Some link types are also systematically filtered out from the exploration: log-out pages, feeds which cannot be opened by the browser, etc.)*

Reload the page

Reloads the current page.

Stop All Processes

Aborts current processes, like the loading of a page, the dig and browse functions, the execution of a macro, etc. In many instances, the escape key has the same effect. *Only active when Outwit is browsing, digging, loading a page, etc*.

Pause All Processes

Pausing complex processes and resuming them at a later time is not always possible. This function gives a simple solution by suspending all processing while displaying an alert and waiting for a click. *Only active when Outwit is browsing, digging, loading a page, etc*.

Bookmarks

Gives access to the bookmarks.

History

Gives access to the navigation history.

Workshop

Loads the 'Workshop Page', a blank page where you can paste and edit any textual content or data to be processed with OutWit Hub.

The Tools Menu

***Gives access to additional tools and features*.**

Available options may vary with the *view* and the license of your product.

Reset All Views

Reverts the settings in the bottom panels of every view to their original values.

Clear History

Clear your browsing history. You can choose to erase everything, or specifically the history of pages you went to, your form filling history, your cache or all your cookies.

Downloads

Opens the download panel.

Preferences

Opens the OutWit Hub's user preference panel.

Apply Scraper

Applies an applicable scraper to the current page.

Apply Macro

Applies a generic macro to the current page.

Error Console

Display the error console with messages (blue), warnings (yellow) and errors (pink) that have occurred recently.

The Help Menu

Gives access to help & support tools and version updates.

Available options may vary with the *view* and the license of your product.

Help

Opens the Help window.

Frequently Asked Questions

Opens a regularly updated list of frequently asked questions.

Tutorials

Opens a list of the built-in Hub tutorials.

Bug Report

Opens the Support Ticket system on outwit.com, at the bug report page.

Suggestions

Opens the Support Ticket system on outwit.com, at the comment & suggestion page.

Version History

Opens the Version History page on outwit.com.

Check for Update

Checks on outwit.com if the current version is the latest. If a new update is available, the program will propose to apply it.

About

Shows the About box, with the version number of the current copy.

The Upgrade / Registration Menu

***To enter and register OutWit Hub Pro license code*.**

Available options may vary with the *view* and the license of your product.

Note: This Menu (the rightmost on the menu bar) reads "Upgrade" in OutWit Hub Light and "Registration" in OutWit Hub Pro.

Enter A Serial Number...

Opens the Registration Dialog to enter email and serial number for unlocking the OutWit Hub Pro features.

Purchase OutWit Hub Pro...

Open the e-shop on outwit.com.

The Datasheet Right-Click Menu

In all datasheets, additional features can be accessed with a right click on the selected items.

Available options can vary with the *view* and the license level of your product.

Edit

Gives access to the Edit sub-menu, with the standard Editing functions and more.

Editing Functions

Cut, Copy and Paste functions are available for cell editing.

Copy Cell(s)

Copies the content of selected cells. Use it to get the contents of selected cells in a column as a list of values.

Paste

(Only active when right-clicking in a directory of the queries view) Pastes the contents of the clipboard as one or two column(s) of items, usually URLs in the first column and some additional info in the second.

Edit Cell

Allows for inline editing of a cell content.

Shuffle Rows

Reorders the datasheet rows randomly. Usually used to avoid sending ordered queries to a server.

Find in Selected Cell(s)...

Opens the find dialog and searches the entered string or regular expression in the selected cells of the current column.

Find in Whole Datasheet...

Opens the find dialog and searches the entered string in the whole datasheet.

Replace in Selected Cell(s)...

Opens the Replace dialog for replacements in the selected cells of the current column.

Replace in Whole Datasheet...

Opens the Replace dialog for replacements in the whole datasheet.

Copy from Column...

Copies values from a column to another.

Use Line as Column Headers

Uses the content of each cell in the selected line to replace the current column headers of the datasheet.

Rename Column...

In data views, this option allows you to change the header of a dynamic column.

Empty Cell(s)

Empties the selected cells of the current column.

Duplicate

Duplicates the contents of selected rows and inserts the duplicates as new rows after the selection.

Insert

Gives access to the Insert and Split sub-menu, with cell/row/column insertion functions.

Insert Row

Inserts a new blank row after the selection.

Insert Rows

Gives access to the String Generation Panel and inserts the generated strings as new rows before the selection. This Insert Rows function allows you to generate strings using the *Query Generation Pattern* format.

Split First/Last Names

If the selected cell values are recognized as people names, this function inserts new 'Fist Name' and 'Last Name' columns before the selected column (if these do not already exist) and fills them with the corresponding values found in the selected cell(s). *Note that, for now, only one pair of First Name/Last Name columns can exist in the datasheet.*

If the 'guess gender' preference of the Filter & Replacements panel contains values, an additional column will be created using these values to indicate the gender associated with the first name. This function based on a large dictionary of first names. When a first name is both used for male and female or when it is not associated to a gender in the dictionary, the program will opt for the third value entered in the preference.

Split Cell(s) to Rows

If the selected cell values contain a character recognized as an item separator (;,-/), this function inserts new rows below the selected rows and fills them with the split values of the selected cells, duplicating the content of the other cells of the selected rows. *Note that, as all 'intelligent' functions, this one can sometimes have unexpected results, but it can nevertheless save you a lot of time in many repetitive tasks.*

Split Cell(s) to Columns

If the selected cell values contain a character recognized as an item separator (;,-/), this function inserts new columns left of the selected column and fills them with the split values of the selected cells. *Note that, as all 'intelligent' functions, this one can sometimes have unexpected results, but it can nevertheless save you a lot of time in many repetitive tasks.*

Insert Column

In data views, inserts a new blank column before the selection. This option only applies to dynamic columns.

Insert Index Column

In data views, inserts a column with an incremeted index. This option only applies to dynamic columns.

Duplicate Column

In data views, inserts a duplicate of the selected column. This option only applies to dynamic columns.

Indexed Duplicate Column

In data views, inserts a duplicate of the selected column where duplicate cells are suffixed with an index. This option only applies to dynamic columns.

Insert Cell(s)

In data views, inserts new blank cells before the selection. This option only applies to dynamic columns.

Delete

Gives access to the Delete sub-menu, to delete cells rows or columns.

Delete

Deletes the selected row(s).

Delete Unselected

Deletes the row(s) that are not selected.

Delete Column

In data views, deletes the selected column. This option only applies to dynamic columns.

Delete Columns

In data views, this options gives you access to a sub-menu allowing you to delete columns containing less than a certain number of populated cells. This option, which only applies to dynamic columns, is very useful to clean up large scrapes where useless columns have been created by poorly populated data fields.

Delete Cell(s)

In data views, deletes selected cells and moves left all the cells located at the right of the selected column. This option only applies to dynamic columns.

Delete Duplicates

Gives access to a sub-menu to delete cell duplicates (rows containing an identical value to the selected cell in the same column) or row duplicates (rows where all cells are identical to the cells of the selected row). It is also possible, through the same menu, to delete all cell or row duplicates of the datasheet.

Select

Gives access to the Select sub-menu, with various ways to select cells or rows.

Select All

Selects all rows of the datasheet.

Invert Selection

Deselects all selected rows of the datasheet and selects all rows that were not selected.

Select Block

In the lists, tables, scraped and news views, this function will select the whole block (list, table, scraped page or rss feed) where the selected row is located. *Note: the selection is done using the second group of digits in the Ordinal ID. (Use the column picker at the top right corner of the datasheet to show the Ordinal column, if it is not visible.)*

Select if in...

In the lists, tables, scraped and news views, this function will select the cells of the selected column that are present in the chosen datasheet. *Note: This function is very useful for instance to check in a query directory which URLs were already scraped with results present i the Catch. Its execution, however, can be very slow with thoudands of items in both datasheets.*

Select Similar

Selects all rows of the datasheet with content similar to that of the the selected cell. The default threshold used for determining similarity is 40 (0 selecting only identical values and 100 selecting everything). Use the sub-menu items to increase or decrease the threshold and select more or less rows.

Select Identical

Selects all rows of the datasheet with content identical to that of the the selected cell.

Select Different

Selects all rows of the datasheet with content different from that of the the selected cell.

Select Duplicates

Gives access to a sub-menu to select cell duplicates (rows containing an identical value to the selected cell in the same column) or row duplicates (rows where all cells are identical to the cells of the selected row). It is also possible, through the same menu, to select all cell or row duplicates of the datasheet.

Auto-Explore

This sub-menu gives access to automation functions that you can apply to the URLs of the selected column in the selected rows of the datasheet. It gives you the capacity to explore the pages or documents and apply extractors, according to the current configuration of the application.

Browse

The program explores the links included in the current selection one after the other. During the exploration, all active extraction processes will be executed on page load, depending on the settings of the views' *bottom panel*.

Browse Selected Links & Series of Pages When Found

The program explores the links included in the current selection one after the other and if a series of pages is found, it will browse through these. During the exploration, all active extraction processes will be executed on page load, depending on the settings of the views' *bottom panel*.

Dig

The program explores the links found in the pages of the current selection's URLs. The exploration will be done within the domain of each link, with a depth of 1. During the Dig process, all active extractions will be executed on page load, according to the settings of the views' *bottom panel*.

Fast Scrape

Applies a Scraper to a list of Selected URLs. When this function is invoked, XML HTTP requests are sent to all the selected URLs, to retrieve the source code of each one. The most relevant scraper is applied to it, without loading images etc. and without any other extraction being performed. All extracted data is sent to the *Scraped* view (which is not emptied during the process, regardless of the state of the *Empty* checkbox).

Fast Scrape (Include Selected Data)

Same function as 'Fast Scrape' above, except that the data fields included in the selection will be added to the scraped results. This saves you the work of merging back the records after the scraping, if you need to keep information from the original data.

Fast Scrape Websites of Selected Links (*Expert & Enterprise editions*)

Same function as 'Fast Scrape' above, except that the scraper will not only be applied to the selected links, but also to all the links of the domain home page that match the filter preference settings defined in the General preference panel.

Apply a Generic Macro

This function allows you to apply a generic macro to the selected URLs. Generic macros are simply macros for which no specific URL is set in the Start Page field.

Fast Search for Contacts

The program sends queries to the site(s) and searches for emails without loading the pages in the browser. Available options in this sub-menu vary with the current page and context. They include:

In Selected Links

The program sends queries to the selected links, searching for email addresses and contact information.

In Websites of Selected Links

The program browses through the pages of the selected links and sends queries to the relevant URLs found (about, contact, team pages) to search for email addresses and contact information.

Open URL in a New Window

In the Firefox Add-on: When the selected data contains a URL, it will be opened in a new browser window.

Download

Gives you access to the Download sub-menu. *Note that a preference (in Tools>Preferences>Export) allows you to automatically rename the downloaded files.*

Download Selected Files

Downloads and saves to the current destination folder on your hard disk, all documents and images found in the selected rows.

Download Selected Files in...

Downloads all documents and images found in the selected rows, opening the folder picker to let you decide where you want the files to be saved.

First Names

Gives you access to the First Names sub-menu.
The First Name Dictionary is used to enhance the recognition of contact in Web pages. A default dictionary of a few thousand first names from around the world is already included in the program. You can add your own using these options. Note that the dictionary can be saved and loaded from the File menu.

Remember First Name

Choosing this option when a first name is selected in the datasheet will add it to your dictionary.

Forget First Name

Choosing this option when a first name is selected in the datasheet will remove it from your dictionary.

Clean Up

Gives access to the Cleaning & Normalization sub-menu.

Clean Contents

Gives access to the String Cleaning sub-menu.

To Lower Case

Converts all characters of the selected cells to lower case.

To Upper Case

Converts all characters of the selected cells to upper case.

Capitalize Words

Converts the first character of each word in the selected cells to Upper case and the others to lower case.

Dust It

Cleans the text at best and capitalizes the words.

Zap It

Cleans the text from all non-alphabeltical chars and capitalizes the words.

Normalize All Figures / Selected Figures in Column

When this function is executed on a selection, the numerical data contained in each selected cell of the selected column (or in the whole datasheet, depending on the selected option) is reformatted and converted to the corresponding value in metric units (if a numerical value is found with a non-metric unit). Numerical values are normalized as much as possible, removing thousand separators, using dots as decimal separators, removing trailing zeros in decimals, etc. The purpose of this function is not to create a nice formatting but rather to homogenize the formats so that the values can be processed and sorted. *Note: the feature is watched by dozens of unit tests in our system and works rather well. There are, however, many possible causes for misinterpretation of numbers in a text, so please do not rely on this function for processes involved in the piloting of commercial airliners, nuclear power plants, etc.*

To Units: Values will be converted to meters, square meters, cubic meters, grams etc.
To k Units: Values will be converted to kilometers, square kilometers, kilograms etc.

Send to Queries

Sends strings to a *directory of Queries*.

Send Cell(s) to Queries

Sends the selected cells to the chosen directory of the queries view:

New Directory: A new directory will be created with the selected items.
directoryName: The selected items will be sent to the chosen directory.

Send Links(s) to Queries

The first links found in the selected rows will be sent to the chosen directory of the queries view:

New Directory: A new directory will be created with the selected items.
directoryName: The selected items will be sent to the chosen directory.

Export Selection as...

Exports the selected data to a file on your hard disk, in one of the available formats (Excel, HTML, Text, CSV, SQL).

The Page Right-Click Menu

In the browser panel, additional features can be accessed with a right click on the page or a click on the Exploration Button.

Available options can vary with the context and the license level of your product.

Edit

Gives access to the Edit sub-menu.

Copy Page Links

If a part of the current page was selected in the browser panel, OutWit Hub will copy the links found in the selection, otherwise all the links of the page will be copied to the clipboard.

Copy External Links

Copies the URLs linking to pages or documents outside the current site's domain.

Copy Local Links

Copies the URLs linking to pages or documents within the current site's domain.

Copy

Copies the selection to the clipboard.

Paste

Pastes the clipboard content.

Paste Text

Pastes the clipboard content as plain text.

Paste Links

Pastes the URLs found in the clipboard content. *Note that if you use this function on the browser, the list of links will replace the currently displayed page.*

Show Last Queries *(Expert & Enterprise editions)*

Displays the last five GET queries that were sent to the server.

Intercept Next Queries *(Expert & Enterprise editions)*

Intercepts and displays the next outgoing queries in two different dialog boxes, one for GET, one for POST. Clicking on OK in these dialogs sends the query to the server. If you do not want to send the query, click on Cancel.

Intercept Next POST Query *(Expert & Enterprise editions)*

Intercepts and displays the next outgoing POST query in the OutWit POST format. Clicking on OK sends the query to the server. If you do not want to send the query, click on Cancel. Displaying the content of the query allows you to save it somewhere (in a directory of the queries view, for instance) and reuse it for automatic form filling, login, etc.

Replace In Next POST Queries (*Expert & Enterprise editions*)

When selected, this menu item opens a dialog which allows you to set one or several string(s) to look for in the outgoing POST query and the corresponding replacement string(s). If you need to replace several strings, separate them using semicolons.
If you enter several lines in the second field of the dialog, the replacement will be done in as many queries as there are lines, allowing you to replace strings in a whole series of outgoing POST queries.

Send Copied Links to Queries

If there are links (URLs) in your clipboard (copied from OutWit Hub or any other application), this function sends them to a new or existing directory of the *queries* view.

Send Highlighted Links to Queries

If hovering over a link, OutWit Hub highlights a series of links it belongs to, this function sends the whole series to a new or existing directory of the *queries* view.

Send Page Links to Queries

Sends all URLs found in the current page to a new or existing directory of the *queries* view.

Edit Page Tools

Gives access to series of functions to alter or reformat the current page (the resulting page can be used as the source for all extractions or saved to your hard disk):

- **Extract All Page Links:** Replaces the currently displayed page with a generated HTML page containing all the links found in this page.
- **Outline Page:**Replaces the currently displayed page with a generated outline of the original page, only keeping the section and paragraph titles and subtitles.
- **Indent Page:**Replaces the currently displayed page with a generated outline of the original page, including the text content, indented within the outline.
- **Decode MIME inclusions:** Replaces MIME inclusions (if any) within the currently displayed page, as legible (and extractable) decoded text.

Select Similar

Selects links that belong to the same series or that are at the same hierarchical level as the selected link.

Select All

Selects all the page content.

Find

Looks for a string or regular expression in the page.

Options

Allows you to disable images, plugins and/or javascript in order to enhance the performance during large automatic explorations. *(These settings are persistent between sessions. Do not forget to switch them back to revert to normal browsing.)*

Fast Search for Contacts

The program sends queries to the site(s) and searches for emails without loading the pages in the browser. Available options in this sub-menu vary with the current page and context. They include:

In Current Website

The program sends queries to the current site and searches for contacts without actually loading the pages in the browser. Not all pages are explored. OutWit Hub tries to locate the ones that are likely to include contact information.

In All Links

The program sends queries to the URLs found in the current page to search for email addresses and contact information.

In Selected Links

The program sends queries to the selected links, searching for email addresses and contact information.

In Highlighted Links

The program sends queries to the highlighted links, searching for email addresses and contact information. *(Hover over the links to highlight series of links.)*

In Linked Websites

The sends queries to the external URLs found (linking outside the current domain) to search for email addresses and contact information.

In Linked Websites (browsing the series of pages)

(Visible if a series of pages was found.) The program browses through the pages of the current series of result pages and sends queries to the external URLs found on these pages (URLs linking outside the current domain), in search for email addresses and contact information.

In All Links (browsing the series of pages)

(Visible if a series of pages was found.) The program browses through the pages of the current series of result pages and sends queries to the URLs found on these pages, in search for email addresses and contact information.

Fast Scrape

Gives access to the Scraper Application sub-menu. Applies the most pertinent scraper to the current page, site, or to the selected / highlighted links. When a scraper is applied to links with this function, *Fast Scrape* mode will be used. Fast Scraping only applies to the page's original source and therefore may not work on dynamic AJAX pages. If you do not want to use the Fast Scrape mode, use the 'Auto-Explore Pages' function instead, after having set the scraped view to recieve the data.
Important: *Note that in Expert & Enterprise editions, the Fast Scrape process will take into account the #addToQueue# directive if used in the scraper, which means that, in this case, self-navigating scrapers can also work in fast mode.*

Current Page

The program sends a query to the current page and applies the most pertinent scraper without actually reloading the page in the browser. (Fast Scraping only applies to the page's original source and therefore may not work on dynamic AJAX pages.)

Current Page Links

The program sends queries to the URLs found in the current page to apply the most pertinent scraper without actually loading the page in the browser. (Fast Scraping only applies to the page's original source and therefore may not work on dynamic AJAX pages.)

Current Site *(Expert & Enterprise editions)*

The program sends queries to the all URLs found in the current page's site, recursively going to all pages of all directories within the domain and applies the most pertinent scraper without actually loading the pages in the browser. (Fast Scraping only applies to the page's original source and therefore may not work on dynamic AJAX pages.)

Linked Websites *(Expert & Enterprise editions)*

The program browses through the pages of the current series of result pages (if any) and sends queries to the external URLs found (linking outside the current domain) to apply the most pertinent scraper without actually loading the page in the browser. All pages of the linked Websites will not be explored. This exploration will scrape the links present on each home page (remaining within each domain). (Fast Scraping only applies to the page's original source and therefore may not work on dynamic AJAX pages.)

This Link, Selected Links, Highlighted Links

The program sends queries to the designated URLs to apply the most pertinent scraper without actually loading the pages in the browser. (Fast Scraping only applies to the page's original source and therefore may not work on dynamic AJAX pages.)

Auto-Explore Pages

Gives access to the Browse/Dig sub-menu. With these functions, the program actually visits and loads each page of a series or selection. Available options in this sub-menu vary with the current page and context.

Browse Selected Links

Auto-browses through the links that are selected in the current page.

Browse Highlighted Links

Auto-browses through the links that are highlighted in the current page. *(Hover over the links to highlight series of links.)*

Browse Series of Result Pages

Auto-browses through the pages of a series.
Active when OutWit finds a navigation link to a following page (i.e. if the current page is part of a series, like a result page for a query in a search engine). Escape or a second click on the Browse button will stop the auto-browse process. Right-clicking or clicking and holding down the Browse button shows a menu allowing to choose the extent of the automatic browse to perform (2,3,5,10 or all pages).

Dig / Browse & Dig Result Pages

Gives access to the Dig sub-menu: The Dig function is a systematic exploration of all links found in a page, in a whole site or in a series of result pages. In order to not visit hundreds of unwanted pages randomly, you can set a number of limitations. You can visit pages if they are within the same domain as the current page, outside the page domain or you can visit any link found. You can also specify the *Depth* of your exploration: Depth 0 is the list of links found in the page, depth 1 also includes all the links found in each visited page and depth 2 does the same one level below. In the *Advanced Settings* dialog, you can combine all these criteria and even set an additional filter with a string (or a *regular expression*) which must be present in the URL for the program to explore it. *(Only links matching the list of extensions set in the Automatic Exploration preference panel are explored in a Dig. Some link types are also systematically filtered out from the exploration: log-out pages, feeds which cannot be opened by the browser, etc.)*

Apply Macro

Applies a generic macro with the current page as start page.

First Names

Gives access to the First Names sub-menu.
The First Name Dictionary is used to enhance the recognition of contact in Web pages. A default dictionary of a few thousand first names from around the world is already included in the program. You can add your own using these options. Note that the dictionary is located in your automator database, which can be saved and loaded from the File menu.

Remember First Name

Choosing this option when a first name is selected in the page will add it to your dictionary.

Forget First Name

Choosing this option when a first name is selected in the page will remove it from your dictionary.

OutWit Hub's Views — *The Side Panel*

***The side panel on the left of your screen contains all available views of the application*.**

The different views allow you to *dissect* the page into its various data elements.
Some display extracted data (links, contacts, text...) others give you access to tools for performing specific extraction tasks (automators).

Items may be collapsed: to display the views they contain, click on the triangle pointed to the right (▶). Some of the sections containing views (like *Data*) are not clickable, as they do not correspond to a view. You need to open the section, if it is collapsed, and select one of the views inside it.

Note: Some views are present in both light and pro versions, with limited or disabled features in the light version, others are only present in the pro version.

▼ page	Displays the current web page or document analyzed in the other widgets.
links	Lists URLs found in the current page.
documents	(Pro) Lists documents found in the current page.
images	Lists images found in the current page.
emails	Lists contact info found in the current page.
▼ data	Contains the data extraction tools.
tables	Extracts HTML table contents.
lists	Extracts HTML list contents.
guess	Tries to guess the structure of the data and extract it.
scraped	Applies the most pertinent active scraper to the page.
▼ text	Displays the current page as simple text.
words	(Pro) Displays the vocabulary used in the page, with the frequency of each word.
news	Displays RSS news found in the current page or domain.
source	Displays the HTML source of the page.
▼ automators	Contains the automation tools.
queries	(Pro) Allows you to create directories of URLs.
scrapers	Allows you to create and edit data scrapers.
macros	(Pro) Allows you to create and edit macros.
jobs	(Pro) Allows you to program the execution of a task.

history	Displays the navigation history, grouped by domain name.

The *Page* View

***This is the browser: it displays the current web page or file that is being analyzed in the other views*.**

When in the *page view*, you can navigate through Web pages as you would in any Web browser. You can also open a local file or even drag a folder from your hard disk to the url bar to see (and navigate through) its content.

Exploration Button and Right-Click Menu

If active, the Explore Button representing a magnifying lens with an at sign (@) is located at the top left corner of the browser. It is the Exploration Button. When moving your cursor across the page, you will see it placing itself above the series of links that the program recognizes and highlights. Automatic navigation functions are available by clicking on this button or by right-clicking directly on the page. *see details in the page right-click menu.*

TIP - Optimizing Performances: *Right-click on the page to disable or reactivate images and plugins in OutWit's browser. Deactivating them can make the loading of each page faster for long explorations and extraction workflows, when you do not need images or flash animations.*

Click on the black triangle next to the *page* view name in the side panel to hide or show the extractors (*links, images, contacts, data, tables, lists, guess, scraper, text, news and source views*).

Note: you can select, in the general preferences, whether you want the application to remain in the current view or to come to this view when a URL is typed in the address bar.

Dragging text to the page

You can Drag a selection from another application to the browser and it will appear as simple text. *This is one of the many ways to import URLs from another source: just drag a selection of urls from a text editor and you will find them in the "links" view. (You can also put them in a .txt file and open the file with the Hub.)*

The *Links* View

***Shows the list of URLs found in the current page*.**

The *links view* displays a table of the URLs found in the current Web page or file, that do not link to media or documents. The table contains the following information:

- **Ordinal**: An index composed of three groups of digits separated by dots*
- **Source URL**: The URL of the page where the link was found*
- **Page URL**: The URL of the link itself
- **Frequency**: The number of occurrences of this link in the page
- **Text**: The description text of the link
- **Filename**: The name of the file the URL links to
- **Type**: The type of document
- **Mime Type**: The Mime Type of the file on the server
- **First Seen**: The first time this link was seen*
- **Last Seen**: The last time this link was seen*
- **Main Doc URL**: The URL of the page's main document. (Useful when a page contains frames or iFrames, to have the parent URL.)*

Note: Columns marked with an asterisk (*) are hidden by default. Use the column picker at the top right corner to show them. When a row is moved to the Catch, the Source URL is always included in the moved data, even if the column is hidden in the view datasheet. All other hidden columns are ignored in the transfer.

If you wish to download some of the files, simply select them in the table and use the '*Download Selected Files*' or '*Download Selected Files in...*' option of the right-click menu.

Bottom Panel Options

When the **local** checkbox is unchecked, OutWit hides links to the same domain as the current page.
When the **cache** checkbox is unchecked, OutWit hides links considered to be cached data.

In general, as for the other views, the content of this view can be filtered and sorted to extract specific data, using the filter controls and right-click functions in the datasheet. You can also move it to the Catch (or export it to a file).

The *Documents* View *(Pro, Expert & Enterprise Editions)*

***Shows the list of all document URLs found in the current page*.**

The documents view displays a table of all document files (.doc, .pdf, .xls, .rtf, .ppt...) found in the file currently displayed in the page view. It includes the following information:

- **Ordinal**: An index composed of three groups of digits separated by dots*
- **Source URL**: The URL of the page where the document was found*
- **Document URL**: The URL of the document
- **Filename**: The name of the file the URL links to
- **Last Modified**: The modification date, if found on the server
- **Size**: The file size
- **Type**: The type of document
- **Mime Type**: The Mime Type of the file on the server

Note: Columns marked with an asterisk (*) are hidden by default. Use the column picker at the top right corner to show them. When a row is moved to the Catch, the Source URL is always included in the moved data, even if the column is hidden in the view datasheet. All other hidden columns are ignored in the transfer.

If you wish to download some of the documents, simply select them in the table and use the '*Download Selected Files*' or '*Download Selected Files in...*' option of the right-click menu.

Bottom Panel Options

In general, as for the other views, the content of this view can be filtered and sorted to extract specific data, using the filter controls and right-click functions in the datasheet. You can also move it to the Catch (or export it to a file).

The *Images* View

***Shows the list of all images found in the current page*.**

The images view displays a table of all image files found in the page currently displayed in the page view or in linked pages.

TIP - Optimizing Performances: *Right-click on the images view name in the side panel to disable or reactivate automatic image extraction when a new page is loaded. Deactivating this can make the processing of each page faster for long explorations and extraction workflows, when you do not need images. (Also see the page view.)*

The table contains the following information:

- **Source URL**: The URL of the page where the image was found*
- **Image**: The thumbnail of the image
- **Filename**: The name of the image file
- **Size**: The size of the image in pixels (width x height)
- **Media URL**: The URL of the image file
- **Found in**: The DOM element where the image was found in the source code (image tag, script, background...)
- **Description**: The size of the image file
- **Type**: The type of image
- **Mime Type**: The Mime Type of the image file on the server
- **Thumb URL**: The URL of the thumbnail (if a high resolution image was found)*

Note: Columns marked with an asterisk (*) are hidden by default. Use the column picker at the top right corner to show them. When a row is moved to the Catch, the Source URL is always included in the moved data, even if the column is hidden in the view datasheet. All other hidden columns are ignored in the transfer.

If you wish to download some of the images, simply select them in the table and use the '*Download Selected Files*' or '*Download Selected Files in...*' option of the right-click menu.

Bottom Panel Options

If the **adjacent** checkbox is checked, OutWit will look for sequences of pictures, by trying to find numerical sequences of in URLs *around* the found images. For instance: if an image named *obama_022.jpg* is found, the program will try to find *obama_021.jpg* and *obama_023.jpg* on the same server.

When the **scripts**, **styles**, **backgrounds**, checkbox are checked, OutWit looks for images in the corresponding tags of the page source code. **Styles** is unchecked by default as style images are often small layout element of lesser interest.

In general, as for the other views, the content of this view can be filtered and sorted to extract specific data, using the filter controls and right-click functions in the datasheet. You can also move it to the Catch (or export it to a file).

The *Contacts* View

***Shows the list of email addresses and contact elements found in the current page*.**

The *emails view* displays a table of the email addresses found in the current Web page / file or in the automatically explored pages. The table contains the following information:

- **Ordinal**: An index composed of three groups of digits separated by dots*
- **Source URL**: The URL of the page where the link was found*
- **Source Domain**: The URL of the page where the link was found
- **Page Title**: The title of the page where the link was found
- **Email**: The email address itself
- **Frequency**: The number of occurrences of this email address in the or in the automatically explored pages
- **Contact Info Columns:** First Name, Last Name, Address, Phone, Fax, Mobile, Toll Free, Title... are added when the 'Guess Contact Info' checkbox is checked in the bottom panel.

Note: Columns marked with an asterisk (*) are hidden by default. Use the column picker at the top right corner to show them. When a row is moved to the Catch, the Source URL is always included in the moved data, even if the column is hidden in the view datasheet. All other hidden columns are ignored in the transfer.

Bottom Panel Options

Strict: When unchecked, OutWit Hub will look for addresses with a looser format and will accept strings like "name at site dot com" or "pseudo[at]domain.org" as valid email addresses.

Guess Contact Info: When checked, OutWit will try to find additional contact information linked to the email address.

Filter Level: Beside the Guess Contact Info checkbox is a popup menu allowing you to select the level of filtering or strictness for the info recognition. When the filter is maximum, the contact data found is only added if it is very likely to be linked to the email address. When the filter is minimum, all found data is added to the result datasheet, at the risk of grabbing some noise or making occasional mistakes associating the info to the email address.

> *Note: The contact info extraction is based on recognition by the program of unstructured data in each page.*
>
> *Recognizing that a series of digits is a phone number rather than a social security number or a date is easy if you know in advance that you are dealing with data from a given country. If you don't, however, the problem is very far from trivial.*
>
> *A brief description of the way OutWit searches for names, addresses, phone and fax numbers etc. will help understand how reliable it can be, depending on the source: The program first looks if additional contact information can be found in the immediate context of each email address. After this, it takes all non-assigned phone numbers and physical addresses and sees if it is likely to belong to one previously found contacts. Otherwise, these are listed independently, lower in the result datasheet.*
>
> *For the data to be extracted, it must first be present in the page, of course. Then, if it is, no technology allows for perfect semantic recognition. An address or a phone number can take so many different forms, depending on the country, on the way it is presented or on how words are abbreviated, that we can never expect to reach a 100% success rate.*
>
> *Email address recognition is nearly perfect in OutWit; phone numbers are recognized rather well in general; physical addresses are more of a challenge: they are better recognized for US, Canada, Australia and European countries than for the rest of the world. The program recognizes names in many cases. As for other fields like the title, for instance, automatic recognition in unstructured data is too complex at this point and results would not be reliable enough for us to include them unless they are clearly labled. We are constantly improving our algorithms so you should make sure to keep your application up-to-date.*

If your need for precision in the extracted data is critical in your workflow and if you cannot afford failed automatic recognition, it may not be a good idea to rely on automatic features like this one. In these cases, you may want to create a scraper for a specific site.

Max Processing Time: Allows you to set the maximum time in seconds that the program should spend analyzing each page when searching for contacts.

Empty/Auto-Empty: This button offers two positions, accessible via the popup arrow on its right side: *Empty on Demand*, which allows you to only clear the contents of the results datasheet when you decide, or *Auto-Empty*, which tells the program to clear the results each time a new page is loaded.

In general, as for the other views, the content of this view can be filtered and sorted to extract specific data, using the filter controls and right-click functions in the datasheet. You can also move it to the Catch (or export it to a file).

The *Data* Section

***Gives access to the different data extraction views of the application*.**

The current version of OutWit Hub Pro, offers four data views: Tables, Lists, Guess and Scraped.

Note: You can hide or show those by clicking on the black triangle next to the section name in the side panel.

The *Tables* and *Lists* views will help you extract data with an explicit structure in the HTML source code of the page. The other data extractors will be useful when these two are not enough to get the Job done. *Guess* tries to automatically recognize the data structure and *Scraped* allows you to manually define how the extraction should be done.

The *Tables* View

Displays the HTML tables found in the current page.

The *tables view* displays in the datasheet, the HTML tables of three rows or more, found in the current page. The minimum number of rows required for tables to be extracted can be altered in the preferences (Tools>Preferences>Advanced Tab).

In case of merged cells in the HTML code, using row or column spans, the cells are kept seperate in the view datasheet and the values will be repeated in the corresponding cells. By default, tables of less than tree rows are ignored. This can be changed in the preferences.

If an hypertext link is found in the data of a table row, it will be placed by the program in the *URL* column at the left of the datasheet. The objective is to gather the useful links in this one column both for the *Lists* and *Tables* views. This column will usually be the simplest way for you to grab collections of links to explore further. If several links are found in each row, outwit will try to decide which column contains the most significant links. By default, the first column containing URLs will be chosen, unless there is a column with less missing links, less duplicate links, etc. This is an arbitrary algorithm, but it usually works pretty well.

The first two columns of the datasheet contain the following information:

- **Ordinal**: An index composed of three groups of digits separated by dots*
- **Source URL**: The URL of the page where the list was found*
- **URL**: The most significant URL found in the table row (if any). Often the first link found.

The following columns vary with the data extracted.

Note: Columns marked with an asterisk (*) are hidden by default. Use the column picker at the top right corner to show/hide them. When a row is moved to the Catch, the Source URL is always included in the moved data, even if the column is hidden in the view datasheet. All other hidden columns are ignored in the transfer.

Bottom Panel Options

In general, as for the other views, the content of this view can be filtered and sorted to extract specific data, using the filter controls and right-click functions in the datasheet. You can also move it to the Catch (or export it to a file).

The *Lists* View

Displays the HTML lists found in the current page.

The *lists view* displays in the datasheet, the HTML lists (<ul>, <ol> and <li> tags) found in the current page, keeping the hierarchical level of the items.

If a link is found for a list item, it will be stored in the *URL* column of the datasheet. If several links are found, only the last one will be kept. The first three columns of the datasheet contain the following information:

- **Ordinal**: An index composed of three groups of digits separated by dots*
- **Source URL**: The URL of the page where the list was found*
- **URL**: The last URL found in the list item (if any)

Note: Columns marked with an asterisk (*) are hidden by default. Use the column picker at the top right corner to show/hide them. When a row is moved to the Catch, the Source URL is always included in the moved data, even if the column is hidden in the view datasheet. All other hidden columns are ignored in the transfer.

Bottom Panel Options

Add Titles: This option was added in v3.0 as lists are often difficult to identify or understand without the title preceeding them. When this option is checked, the program includes the content of <title> and <h1> <h2> <h3>... tags in the HTML page.

In general, as for the other views, the content of this view can be filtered and sorted to extract specific data, using the filter controls and right-click functions in the datasheet. You can also move it to the Catch (or export it to a file).

The *Guess* View

Displays the data extracted using automatic structure recognition algorithms.

The *guess* view tries to understand the structure of the data found in the current page, if any.

> *Note: The program analyzes the available html source code of the page. Labels and field/record separators are looked for, using many different strategies. The program eventually gives a rating to each possible structure found and decides of the best possible answer, if any. The Challenge of these intelligent algorithms is to understand even non-tabulated data and we will make sure they become more and more efficient, but the very nature of the problem makes it impossible to ever get close to a 100% success rate.*
>
> *If your need for the scraped data is critical in your workflow and if you cannot afford failed automatic recognition, it may not be a good idea to rely on automatic features like this one. In these cases, you should probably define a scraper and use the right click menu option: 'Auto-Explore' > 'Fast Scrape'. This way, if you have thoroughly tested the scraper you have designed, the process will be reliable and reproducible, at least as long as the online source is not altered and remains accessible.*

Ordinal: An index composed of three groups of digits separated by dots (hidden by default)

Source URL: As in the datasheets of the other views, the *Source URL* is placed in the second column and is hidden by default.

Note: Use the column picker at the top right corner to show/hide them. When a row is moved to the Catch, the Source URL is always included in the moved data, even if the column is hidden in the view datasheet. All other hidden columns are ignored in the transfer.

Bottom Panel Options

List: When checked, OutWit will try to find a list of records and present them as a table *(one record per row, one field per column)*. When unchecked, the program will try to recognize *"specsheet"* type of data in the page *(one row per field, a Label and a Value for each field)*. If you uncheck this option, OutWit should do better with simple text data like this:

last name: Knapp
first name: John
age: 34
phone: (674) 555-5621

You can try this option by going to the *workshop* page (ctrl/cmd-shift-k) and pasting text from a word processor, emails... Guess will usually do better if you paste simple text, using a right-click and chosing Edit>Paste Text.

In general, as for the other views, the content of this view can be filtered and sorted to extract specific data, using the filter controls and right-click functions in the datasheet. You can also move it to the Catch (or export it to a file).

The *Scraped Data* View

Displays the results of the application of a scraper to the current page (or to a series of URLs).

The *scraped data* view displays in a table the data extracted using the active **scraper** with the highest rating. Each field defined in the scraper corresponds to a column of the datasheet.

If several active scrapers can be applied to the current URL, the possible candidates will be rated according to their version number, their freshness and the specificity of the *Apply to URL* (*mySite.com having a lower priority than www.mySite.com/myPage*). If you wish to apply a scraper of a lesser rating, you can deactivate all scrapers with a higher priority in the *scraper manager*.

Ordinal: An index composed of three groups of digits separated by dots
Source URL: As in the datasheets of the other views, the *Source URL* is placed in the first column and is hidden by default.

Note: Use the column picker at the top right corner to show/hide them. When a row is moved to the Catch, the Source URL is always included in the moved data, even if the column is hidden in the view datasheet. All other hidden columns are ignored in the transfer.

Bottom Panel Options

The **Keep Order** checkbox allows you to ask OutWit to force the columns in the same order as the scraper. If this option is checked, all columns will appear in the resulting data, even when they are completely empty. (This can be useful if you wish to export to an Excel or HTML file, for instance as part of a job, to then use this data in a set process, with other applications.)

In case of application of a scraper to whole lists of URLs, like with the 'Auto-Explore' > 'Fast Scrape' of the datasheets' right-click menu, the **Empty** checkbox will be ignored. In all other cases, if this option is checked, the datasheet will be emptied as soon as a new page is loaded.

In general, as for the other views, the content of this view can be filtered and sorted to extract specific data, using the filter controls and right-click functions in the datasheet. You can also move it to the Catch (or export it to a file).

The *Text* View

***Shows the current page as simple text*.**

The *Text view* displays the textual content of the current page and ignores all other content: scripts, media, animations, layout, etc.

Note: You can hide or show the Text related views available in your version of OutWit Hub, by clicking on the black triangle next to the view name in the side panel.

All or parts of the text can be moved to the Catch (or saved to a file).

The *Words* View *(Pro, Expert & Enterprise Editions)*

***Displays the vocabulary used in the page, with the frequency of each word*.**

The *Words* view displays a table of significant words and groups of words found in the source code of the current Web page or file. The frequency column gives you, as a fraction, the number of occurrences divided by the total number of words. Note that if you notice a higher number of occurrences than what you can actually see in the Web page, it means that the other occurrences of the word or phrase are in the source code but hidden (like alternate text, invisible blocks, etc.).

Groups of words are recurring successions of two to four words. If OutWit recognizes the page language, "empty words" are ignored in this view. This means that in English, French, German, Spanish and several other occidental languages, very common pronouns, auxiliaries, articles, etc. will be ignored. This covers words like "the", "is", "which" etc.

The table contains the following information:

- **Ordinal**: An index composed of three groups of digits separated by dots*
- **Source URL**: The URL of the page where the word was found *(or the number of pages where it appeared, if 'Empty' is unchecked)*
- **Word**: The word
- **Frequency**: The number of occurrences of this word in the page

In the *Expert & Enterprise editions* , a text box is present below the datasheet. You can type words or paste a text in this box to count only specific words in the page.

Note: The Ordinal column is hidden by default in this view. Use the column picker at the top right corner to show/hide them. When a row is moved to the Catch, the Source URL is always included in the moved data, even if the column is hidden in the view datasheet. All other hidden columns are ignored in the transfer.

Bottom Panel Options

In general, as for the other views, the content of this view can be filtered and sorted to extract specific data, using the filter controls and right-click functions in the datasheet. You can also move it to the Catch (or export it to a file).

The *News* View

***Shows the list of RSS articles found in the current page or in the current domain*.**

The *news view* displays a table of the news articles from all RSS feed found in the current Web page, or in the same domain. The table contains the following information:

- **Ordinal**: An index composed of three groups of digits separated by dots*
- **Source URL**: The URL of the page where the feed was found*
- **Feed URL**: The URL of the RSS feed*
- **Feed Title**: The name of the feed*
- **Feed Link**: The link of the HTML page corresponding to the feed*
- **Feed Description**: The description of the feed*
- **Feed Language**: The language of the RSS feed*
- **Title**: The title of the article
- **Article URL**: The link to the full article
- **Date**: The date and time of release
- **Image**: The URL to the attached image*
- **Category**: The category name of the article*
- **Abstract**: The abstract of the article

Note: Columns marked with an asterisk (*) are hidden by default. Use the column picker at the top right corner to show them. When a row is moved to the Catch, the Source URL is always included in the moved data, even if the column is hidden in the view datasheet. All other hidden columns are ignored in the transfer.

Bottom Panel Options

In general, as for the other views, the content of this view can be filtered and sorted to extract specific data, using the filter controls and right-click functions in the datasheet. You can also move it to the Catch (or export it to a file).

The *Source* View

***Shows the colorized source code of the current page*.**

When the current file displayed in the *Page* view is a Web page, the *source view* contains the colorized HTML source code of its main document.

The light version displays the **original** source code, as it was loaded by the browser when opening the page. In the pro version, a popup menu allows you to select if you want to display and scrape the **original** source code or the **dynamic** source code, altered by scripts after the page was loaded. In addition, the *Expert and Enterprise* editions can display the dynamic source as a concatenation of **all the frames** composing the page.

The *dynamic* source code is presented on a pale yellow background, the *all frames* version, on a light green backgroung and the *original*, on a white background. This will help you recognize the setting immediately.

The source code presented in the scrapers view is another instance of the same panel.

The colorization was conceived for data search rather than programming purposes and emphasis is given to the textual content that is actually displayed on the page: it is shown in black and pops out from the cryptic HTML syntax.

The colors used are the following:

- **Displayed text**
- **HTML tags**
- **Links**
- **Comments**
- **HTML Entities**
- **Styles**
- **Scripts**
- **Images**

The *History* View

Displays the list of seen URLs grouped by domain.

The *history* view doesn't show the list of URLs that have been *visited* (this would be redundant with the browser navigation history), but of URLs that have been *seen* in the current session. This means that, as the history is grouped by domain, after surfing for 15 minutes (or hours) on Web pages related to a certain topic, say astronomy, you will find in this view a list of the most frequently cited domains in this topic.

The *History* datasheet contains the following information:

- **Domain**: The domain
- **First Seen on**: The time when this domain was first recorded in the current session
- **Last Seen on**: The time when this domain was seen most recently in the current session
- **Frequency**: The number of occurrences of this link in the page

Bottom Panel Options

In general, as for the other views, the content of this view can be filtered and sorted to extract specific data, using the right-click functions in the datasheet. You can also move it to the Catch (or export it to a file).

The *Automators* Section

***Gives access to the different automators available in the application*.**

In the current version of OutWit Hub Pro, five kinds of automators can be defined: Projects, Scrapers, Macros, Jobs and Queries.

- A **Scraper** is a description of the data structure in a page, defining the markers that can be found in the document source code, around the data you wish to extract.
- A **Macro** is a snapshot of the complete configuration of the application’s various extractors, which can be *replayed* in a single click for performing a specific exploration and extraction task.
- A **Job** is a preset time and periodicity at which an action should be performed.
- A Set of **Queries** is a directory containing a list of URLs or Query Matrices on which an action (autobrowsing, macro, scraper, slideshow, etc) can be performed.
- A **Project** is a directory containing a list of automators (Scrapers, Macros, Queries, Jobs).

The *Automator Managers*

Allows you to manage the automators stored in your profile.

In each of the automator views (Projects, Scrapers, Macros, Jobs and Queries), the manager is the panel presenting the list of all automators of the considered type stored in your profile. The manager allows you to create, delete, import, export automators and gives access to the property and automator editors.

Each automator is identified by its Automator ID (AID) in your profile. It is preceded by an **Active** checkbox. When this box is unchecked the automator is deactivated and grayed out in the list. You will need to activate it before using it.

If a layout change button is present at the top right corner of the panel, you will be able to switch between horizontal and vertical layout of the window, placing the editor and manager at the bottom or on the right of the screen.

Right-clicking on the name of an automator in the manager, allows you to:

- **Delete** Deletes the current automator from your profile.
- **Duplicate** Duplicates the current automator.
- **Export** Exports the automator as an XML file or as a User Gear file (.owc) to your hard disk.
- **Properties** Displays the Properties Dialog where information about the current automator can be found and edited (name, author, comments, etc.).
- **Execute** Executes the automator (if applicable).
- **New** Opens a new blank automator.
- **Import** Imports a previously exported automator.
- **Assign to Project** (*Expert & Enterprise editions*) Adds the selected automator to a new or existing project.

Sharing automators (*Enterprise edition v7.0+*)

With the Enterprise edition of OutWit Hub v7.0+, you can share an automator database between several users or several instances of the application on a local hard disk, a local network or over the Internet. The sharing configuration settings are located in the Advanced Preference Panel.

These are the possible database sharing settings:

- **Sharing OFF:** [DEFAULT SETTING] The automator database used is the one in your own profile. You have read and write access to it.
- **Sharing ON, Local Network Address, Write Access OFF, Import OFF:** At startup, the application loads the shared database from the path you entered. You will not be able to create, modify or rename automators. The automator manager datasheet background color is a light gray instead of green, to remind you of the setting.
- **Sharing ON, Local Network Address, Write Access OFF, Import ON:** At startup, the application imports all automators from the database at the path you entered into the one in your own profile. The automator database used after this is the one in your profile. You have read and write access to it. The automator manager datasheet background color is the default light green and imported automators are displayed with a blue text. IMPORTANT: These imported automators will be overwritten during the next startup of the application. If you decide to modify them, the version number will need to be changed or your changes will be lost.
- **Sharing ON, Local Network Address, Write Access ON, Import OFF:** The automator database used is the shared database at the path you entered. You will be able to create, modify or rename automators and these will alter the shared database. YOUR CHANGES WILL IMPACT EVERYONE USING THE DATABASE. The automator manager datasheet background color is a light pink instead of green, to remind you of the setting.
- **Sharing ON, Web Address, Write Access OFF, Import OFF:** At startup, the application loads the shared database from the Web URL you entered. You will not be able to create, modify or rename automators. The automator manager datasheet background color is a light gray instead of green, to remind you of the setting.
- **Sharing ON, Web Address, Write Access OFF, Import ON:** At startup, the application imports all automators from the database at the Web URL you entered into the one in your

own profile. The automator database used after this is the one in your profile. You have read and write access to it. The automator manager datasheet background color is the default light green and imported automators are displayed with a blue text. IMPORTANT: These imported automators will be overwritten during the next startup of the application. If you decide to modify them, the version number will need to be changed or your changes will be lost.

The *Automator Managers*

***Allows you to manage the automators stored in your profile*.**

In each of the automator views (Projects, Scrapers, Macros, Jobs and Queries), the manager is the panel presenting the list of all automators of the considered type stored in your profile. The manager allows you to create, delete, import, export automators and gives access to the property and automator editors.

Each automator is identified by its Automator ID (AID) in your profile. It is preceded by an **Active** checkbox. When this box is unchecked the automator is deactivated and grayed out in the list. You will need to activate it before using it.

If a layout change button is present at the top right corner of the panel, you will be able to switch between horizontal and vertical layout of the window, placing the editor and manager at the bottom or on the right of the screen.

Right-clicking on the name of an automator in the manager, allows you to:

- **Delete** Deletes the current automator from your profile.
- **Duplicate** Duplicates the current automator.
- **Export** Exports the automator as an XML file or as a User Gear file (.owc) to your hard disk.
- **Properties** Displays the Properties Dialog where information about the current automator can be found and edited (name, author, comments, etc.).
- **Execute** Executes the automator (if applicable).
- **New** Opens a new blank automator.
- **Import** Imports a previously exported automator.
- **Assign to Project** Adds the selected automator to a new or existing project.

Sharing automators (*Enterprise edition v6.0+*)

With the Enterprise edition of OutWit Hub v6.0+, you can share an automator database between several users or several instances of the application on a local hard disk, a local network or over the Internet. The sharing configuration settings are located in the Advanced Preference Panel.
The first application of this function is to use a common set of automators for several instances of the program. Sharing a small automator database between several users, in read-only mode on a local network, is also a good application. Note that if the database contains more than a few scrapers, macros or jobs, overall performance can become a problem, especially, of course, if the network is not very fast. The possibility to edit the shared database remotely is offered for the sake of completeness but it is not recommended both for perfomance and security reasons.

How to use a shared automator database:
If the shared database (let's call it Shared_Gear.owg) is located on a local hard disk, simply put the path of the file in the textbox of the Advanced Preference Panel (something like C:\Users\...\Shared_Gear.owg on Windows, /Users/.../Shared_Gear.owg on Macintosh or /home/.../Shared_Gear.owg on Linux machines). If the database is located on a shared network drive, make sure that the network is fast enough and that your database is not too large or performances could be a problem:

- **On a Macintosh:** Mount the network drive (Go>Connect...), create an alias of the distant file, select it and look at the informations (cmd-I), it will show you, with the label 'Original:', the path that should be used in the advanced preference textbox: something like /Volumes/.../Shared_Gear.owg.
- **On Windows:** Mount the network drive, select the shared database file and look at the properties (usually ALT+ENTER), it will show you, with the label 'Location:', the path to the folder that contains your file. Add the file name to get the path that should be used in the advanced preference textbox: something like E:\...\Shared_Gear.owg.
- **On Linux:** Mount the network drive, create a link to the distant file, select it and look at the properies, it will show you, with the label 'Location:', the path that should be used in the advanced preference textbox.

These are the possible database sharing settings:

- **Sharing is OFF:** [DEFAULT SETTING] The automator database used is the one in your own profile. You have read and write access to it.
- **Sharing is ON on a Local Network Address, Write Access OFF, Import OFF:** At startup, the application loads the shared database from the path you entered. You will not be able to create, modify or rename automators. The automator manager datasheet background color is a light gray instead of green, to remind you of the setting.
- **Sharing is ON on a Local Network Address, Write Access OFF, Import ON:** At startup, the application imports all automators from the database at the path you entered into the one in your own profile. The automator database used after this is the one in your profile. You have read and write access to it. The automator manager datasheet background color is the default light green and imported automators are displayed with a blue text. IMPORTANT: These imported automators will be overwritten during the next startup of the application. If you decide to modify them, the version number will need to be changed or your changes will be lost.
- **Sharing is ON on a Local Network Address, Write Access ON, Import OFF:** [NOT RECOMMENDED] The automator database used is the shared database at the path you entered. You will be able to create, modify or rename automators and these will alter the shared database. YOUR CHANGES WILL IMPACT EVERYONE USING THE DATABASE. The automator manager datasheet background color is a light pink instead of green, to remind you to be cautious with this setting.
- **Sharing is ON on a Web Address, Write Access OFF, Import OFF:** At startup, the application loads the shared database from the Web URL you entered. You will not be able to create, modify or rename automators. The automator manager datasheet background color is a light gray instead of green, to remind you of the setting.
- **Sharing is ON on a Web Address, Write Access OFF, Import ON:** At startup, the application imports all automators from the database at the Web URL you entered into the one in your own profile. The automator database used after this is the one in your profile. You have read and write access to it. The automator manager datasheet background color is the default light green and imported automators are displayed with a blue text. IMPORTANT: These imported automators will be overwritten during the next startup of the application. If you decide to modify them, the version number will need to be changed or your changes will be lost.

Important Notes:

1) *If you are sharing an automator database on a local area newtork and have writing privileges, a backup of the shared database will be done when the application is launched. If the network is not very fast and if the database is big, the process can be very long. It is a better idea to do your modifications in the local User_Gear.owc database (see Help>Standalone Application for info on the location of the files) then place this database in the shared folder for others to use it.*

2) *If you are the system administrator and wish to prevent any other person from getting writing privileges in the application preferences for a shared database, please contact OutWit's support on outwit.com with the credentials that were used to purchase the program and ask us for the settings.*

The *Scrapers* View

***A Scraper is a template telling OutWit how to extract information from a page*.**

When '**tables**', '**lists**' or '**guess**' do not manage to recognize automatically the structure of a page and extract its data, you still have the option to create a *Scraper* and tell OutWit how it should handle this specific URL (or all the pages of a given Web site, a sub-section thereof, etc).

OutWit Hub offers a whole list of **directives and functions to design and customize your scrapers**.

What is a scraper?

A scraper is simply a list of the fields you want to recognize and extract. For each field, it specifies the name of the field (ex.: 'Phone Number'), the strings located immediately before and after the data to extract in the source code of the page and the format of the data to extract for this field. The pro version also allows you to set replacement string to alter the extracted data and a delimiter, to split the extracted result into several fields.

Creating and Editing Scrapers

The *scrapers* view can contain either the scraper manager (to create, duplicate, delete previously made scrapers), or the **scraper editor**, to build and edit them.

In editing mode, the source code of the current page is displayed, for you to easily identify and copy the markers you need. You can select the source code you want your scraper to be applied to: the **original** source code (white background) on all versions of OutWit Hub, the **Dynamic** source code (pale yellow background), if you are using the Pro version and **All Frames** (pale green background) if you are using the Expert & Enterprise editions. To select the source, use the *source type* popup menu.
In the bottom part of the window is the editor itself, where you can create and modify your scrapers.

To switch from one mode to the other, use the Manage or Edit button.

Activating scrapers

As for all other extractors, the 'scraped' view is automatically active when any control of the bottom panel is set to a non-default value. (i.e. 'empty' is unchecked, 'move to catch' is checked...). On the scrapers themselves, an additional control must be set for the extraction to be done automatically:

In the scraper manager, the 'active' checkboxes determine whether a scraper should be used when the corresponding URL is loaded. When unchecked, the scraper is deactivated.

When asked to scrape a page, if there are more than one applicable active scrapers for that URL, OutWit will apply the one that seems most appropriate and recent, using several criteria (version number, modification time, specificity of the 'URL contains' string, etc.).

In the light version, only one scraper can be active at a given time and no more than ten scrapers can be present in the manager.

Scraping Data

- If a control of the bottom panel is set to a non-default value in the **scraped** view, the program will try to find a matching scraper and apply it, as soon as a new page is loaded.
- From both the manager and the editor, you can apply the scraper to the current page, using the 'Execute' button (or the right-click menu, in the manager). After performing an extraction with a scraper, you will find your results in the scraped data view.
- From any datasheet, you can select URLs, right click on one of them and choose **'Auto-Explore' > 'Fast Scrape'** in the popup menu.
- You can of course include your scrapers in **macros** (browsing/digging through URLs or using the 'fast scraping' mode) for recurring extraction tasks.

The *Scraper* Editor *(OutWit Scrapers Syntax Reference)*

In the *scrapers view*, the bottom part of the window can either be the *Scraper Manager* or the *Scraper Editor*. In the scraper manager, you can see and organize your scrapers and, when double-clicking on one of them or creating a new one using the **New** button, the scraper editor opens and you can create or alter your scraper lines.

At the bottom of the editor, the following series of buttons gives you access to the general management functions:

- **Save** Saves the current scraper to your profile
- **New** Opens a new blank scraper
- **Duplicate** Duplicates the selected lines
- **Delete** Deletes the current scraper from your profile
- **Export** Saves the scraper as an XML file to your hard disk
- **Import** Loads a previously exported scraper
- **Revert** Restores the last saved version of the scraper
- **Properties** Displays the Scraper Properties Dialog where information about the current macro can be found and edited (name, author, comments, etc.)
- **Close** Closes the Scraper Editor and displays the Scraper Manager.

The Editor itself allows you to define the following information:

Apply if URL contains...: The URL to Scrape (or a part thereof). This is the condition to apply the scraper to a page. The string you enter in this field can be a whole URL, a part of URL, or a regular expression, starting and ending with a '/'. (In the last case, the string will be displayed in red if the syntax is invalid.) If you try to scrape a page with the 'Execute' button when this field doesn't match the URL, an error message will be displayed. If two or more scrapers match the URL of the page to be scraped, the priority will be given to the most recent, with the most significant condition (longest match).

> *Note: If you keep getting the message: **"This scraper is not destined to the current URL"**, this is the field that must be changed. A frequent mistake is to put a whole URL in this field, when the scraper is destined to several pages. Try to enter only the part of the URL which is common to all the pages you wish to scrape, but specific enough to not match unwanted pages.*
>
> *You may also get this error message if you are trying to apply a disabled scraper to a valid URL. In this case, just check the OK checkbox in the scraper manager.*

Source Type *(Pro, Expert & Enterprise Editions)*: You can set your scraper to be applied either to the original source code that was loaded by the browser when opening the page or to the code as it was dynamically altered by scripts after the page was loaded.
In the *Expert & Enterprise editions* you can also select the dynamic source as a concatenation of all the frames composing the page.

In the scraper definition itself, each line corresponds to a field of data to be extracted. *(If you leave the scraper empty, its **default behavior will be to scrape the whole text of the page**, removing HTML tags and styles and keeping line breaks and basic layout when possible.)*

***To edit** or enter a value in a cell, **double-click on the cell**. To simplify the fabrication of scrapers and avoid typos, the best way is often to select the string you want in the source code and drag it to the cell. You can then edit it as you like.*

- **Description** (name of the field): can either contain a simple label like *"Phone"* or *"First Name"* or a *directive* (see below),
- **A) Marker Before** - *optional*: a string or a Regular Expression marking the beginning of the data to extract,
- **B) Marker After** - *optional*: a string or a Regular Expression marking the end of the data to extract,
- **C) Format** - *optional*: a Regular Expression describing the format of the data to extract,
- **Replace** - *(pro version) optional*: replacement pattern (or value to which this field must be set).

- **Separator***(pro version) optional*: delimiter, to split the extracted result into several fields.
- **List of Labels***(pro version) optional*: the list of labels to be used, if the result is split into several fields with a separator.

Important Notes:

*1) In a scraper, a line doesn't have to include Marker Before (A), Marker After (B) and Format (C). One or two of these fields can be empty. The **authorized combinations** are: ABC, AC, BC, AB, A, C.*

*2) When creating a scraper you can **right-click on a marker or format field** and choose **Find in Source** to find and highlight the occurrences of a string or a pattern in the source code of the current page. If you right-click on the description field it will allow you to find the whole scraper line in the source code. This is very useful for troubleshooting.*

*3) **Record Separator:** The first line of the scraper will be considered by OutWit Hub as the field that starts a new record. This means that each time this scraper line matches data in the page, a new record will be created. Usually, the best way is to follow the order of appearance of the fields in the source document.*

In the **Format** pattern, use the regular expression syntax and do not forget to escape reserved characters.

> *Note: If you right-click on the text you have entered in a cell, an option will allow you to escape a literal string easily. In the Format field, the content will always be understood as a regular expression, even if not surrounded by **/ /**.*

In the **Replace** string, use **\0** to insert the whole string extracted by this line of the scraper, or **\1**, **\2**, etc. to include the data captured by parentheses --if any-- in the *Format* regular expression.

For instance, say you extract the string "0987654321" with a given scraper line. Adding a replacement pattern can help you rebuild a whole URL from the extracted data:

If you enter:
http://www.mySite.com/play?id=\0&autostart=true
as replacement string, the scraper line will return
http://www.mySite.com/play?id=0987654321&autostart=true

In the **Separator**, use either a literal string like **,** or **;** or a regular expression like **/ [,;_\-\/] /**.

For technical reasons, all regular expressions used in scrapers are interpreted as case insensitive patterns by default. [A-Z], [A-Za-z] and [a-z] have the same result. This can be changed using the #caseSensitive# directive. This means that 'Marker before', 'Marker after', 'Format', which are always converted to regular expressions by the program, are case insensitive by default. Conversely, the 'Separator 'which is used *as is* by the program, is therefore case sensitive by default if it is a literal string, and case insensitive if it was entered as a regular expression.

When splitting the result with a separator, use the **List of Labels** to assign a field name to each part of the data. Separate the labels with a comma. If there are less labels than split elements or if the Labels field is empty, default labels will be assigned using the description and an index.

Example: the string you want to extract doesn't have remarkable markers and you do not know how to separate different elements of the data. Say the source code looks like this:

<li>Dimensions:35x40x70</li>

If you want the three dimensions in three separate columns, you can reconstitute the structure by entering the following:

Marker Before:
<li>Dimensions:
Marker After:
</li>
Separator:
x
Labels:
Height,Width,Depth

Regular Expressions in Your Scraper

To learn about Regular Expressions, please visit the [RegExp Quick Start Guide]

Regular expressions can be used in several parts of the scraper:

- Apply if URL contains
- Marker Before
- Marker After
- Format
- Separator

They allow you to keep the scraper short if you are confortable with them. In many cases, however, it is also possible to do without. For instance, if you need a OR, just create two scraper lines with the same field name (Description).

Example: If you want your scraper to match both 'sedan-4 doors' and 'coupe-2 doors'
The simple way is do it in two separate lines:
Description:
car
Format:
sedan-4 doors
Description:
car
Format:
coupe-2 doors

Or you can use a regular expression:
Description:
car
Format:
/(sedan\-4|coupe\-2) doors/

Directives *(Pro, Expert & Enterprise Editions)*

Directives alter the normal behavior of the scraper. They can be located anywhere in the scraper and will be interpreted before all other lines by the program. Directives are identified by # characters in the description field:

Alphabetical list of available directives (Click on one for details):

- **abortAfter** aborts the current extraction after the current page, if the scraper line matches.
- **abortAfterNPages** *(Expert & Enterprise)* aborts the current extraction after the Nth explored page.
- **abortAfterNResults** *(Expert & Enterprise)* aborts the current extraction when the scraped view contains N extracted records.
- **abortIf** aborts the current extraction immediately if the scraper line matches.
- **abortIfNot** aborts the current extraction immediately if the scraper line doesn't match.
- **addToQueue** stores the data scraped by the line in the queue.
- **addURLToSource** *(Expert & Enterprise)* adds the URL at the beginning of the source code
- **alertOnStop** displays an alert at the end of the exploration process.
- **allFrames** *(Expert & Enterprise)* scrapes the concatenation of all frames in the current page.
- **allowCrossDomain** *(Expert & Enterprise)* removes javascript restrictions and reloads the page.
- **autoCatch** *(Expert & Enterprise)* forces the position of the Empty/Catch option of the scraped view to On Demand or Auto.
- **autoEmpty, emptyOnDemand** *(Expert & Enterprise)* set the value of the 'Empty' checkbox.
- **caseSensitive** makes the whole scraper case sensitive.
- **catchEvery,catchAndDeleteEvery** *(Expert & Enterprise)* sends scraped data to the Catch every n collected rows.
- **catchOnStop** *(Expert & Enterprise)* sends scraped data to the Catch at the end of the exploration.
- **checkIf** and **checkIfNot** condition the activation of the passed lines on page content.
- **checkIfURL** and **checkIfNotURL** conditions the activation of the passed lines on URL.
- **cleanData** and **originalData** clean the extracted data or not
- **cleanHTML** normalizes the HTML tags before the scrape.
- **clearAllHistory** *(Expert & Enterprise)* clears all history, forms, cookies, cache from the browser memory.
- **clearBrowsingHistory** *(Expert & Enterprise)* clears browsing history.
- **clearCookie** *(Expert & Enterprise)* clears cookies stored for the current URL.
- **clearCookieEvery** *(Expert & Enterprise)* clears cookies for current URL every n pages.
- **clearCookieIf** *(Expert & Enterprise)* clears cookies stored for the current URL if the scraper line matches.
- **clearCookiesEvery** *(Expert & Enterprise)* clears all cookies every n pages.
- **clearCookiesIf** *(Expert & Enterprise)* clears all cookies if the scraper line matches.
- **clearCookiesIfNot** *(Expert & Enterprise)* clears all cookies if the scraper line does not match.
- **clearForms** *(Expert & Enterprise)* clears form data from the browser memory.
- **coalesceOnStop** *(Expert & Enterprise)* coalesces extracted data keeping the first value in each column.
- **concatSeparator** sets the delimiter to use for concatenation.
- **decodeEntities** *(Expert & Enterprise)* Decodes HTML entities (like &amp; or &gt;) to their plain text equivalent.
- **decodeJSCharcodes**Converts hexadecimal and decimal codes into characters.
- **deduplicate** *(Expert & Enterprise)* removes duplicates from extracted data.
- **deduplicateOnStop** *(Expert & Enterprise)* removes duplicates from extracted data after exploration.
- **deduplicateWithinPage** *(Expert & Enterprise)* does a row by row deduplication for each scraped page.
- **default** *(Expert & Enterprise)* sets default value for *myFieldName*.
- **deleteCookies, deleteCookiesIf, deleteCookiesIfNot** *(Expert & Enterprise)* deletes cookies.
- **download** *(Expert & Enterprise)* downloads file at URL grabbed by scraper line.
- **downloadReferer** *(Expert & Enterprise)* sets the default referer for the next URL(s) downloaded.
- **emptyDirectory** *(Expert & Enterprise)* Empties the first directory of the queries view matching the passed name.
- **emptyOnDemand** forces the position of the Empty option of the scraped view to On Demand or Auto.
- **exclude** ignores a field if it contains this value
- **excludeFromQueueIf** *(Expert & Enterprise)* excludes urls that match a string or a regular expression.

- **excludeFromQueueIfNot** *(Expert & Enterprise)* excludes urls that do not match a string or a regular expression.
- **exportEvery,exportAndDeleteEvery** *(Expert & Enterprise)* exports, every n collected rows, the data stored in the scraped view.
- **extractContacts** *(Expert & Enterprise)* applies the contact extractor to the current URL loaded in the browser.
- **fieldGroup** *(Expert & Enterprise)* makes sure field indexes of a same group are incremented together.
- **getJSON** *(Expert & Enterprise)* ignores all other lines and tries to interpret JSON blocks.
- **getLocalIP** gets public IP address of the machine from the network.
- **hideNodes** *(Expert & Enterprise)* Makes the nodes matching the passed css selector invisible.
- **holdResults, holdResultsIf** and **holdResultsIfNot** *(Expert & Enterprise)* hold (to later merge) data scraped from different pages
- **ignoreErrors** Cells where a function returned an error are empty.
- **indentedText** indents the document/page before scraping.
- **insertIf** inserts the content of the replace cell if this scraper line matches.
- **insertIfNoResults** adds a line with the content of the replacement column if no data was extracted in the page.
- **insertIfNot** inserts the content of the replace cell if this scraper line doesn't match.
- **insertRow** *(Expert & Enterprise)* inserts a row with the value of the replacement column for every record of the final output.
- **keepForms** does not remove forms when cleaning the extracted text.
- **keepOrder** has the same effect as checking the 'keep order' checkbox.
- **limit** *(Expert & Enterprise)* only keeps the n first rows of extracted data.
- **maxColumn** *(Expert & Enterprise)* sets the maximum number of columns in the extracted datasheet.
- **maxIndex** *(Expert & Enterprise)* sets the maximum number of columns for a given field.
- **mergeResults, mergeResultsIf** and **mergeResultsIfNot** *(Expert & Enterprise)* concatenate previously stored data with the data scraped in this page.
- **minIndex** *(Expert & Enterprise)* sets the minimum number of columns with the same name.
- **newRecord** creates a new record (new row) for each match .
- **nextPage** sets the link of the next page to use in an automatic browse process
- **nextPageMax** *(Expert & Enterprise)* sets the maximum number of pages to be explored.
- **nextPageReferer** *(Expert & Enterprise)* sets the referer for the next page query that will be sent to the server.
- **normalizeToK** and **normalizeToUnits** normalize numerical value in the field to decimal units or k units
- **oneRow** present all page data as a single row.
- **originalData** does not clean the scraper result cells.
- **originalHTML** scrapes the original HTML source without any alteration.
- **outline** alters the source code before scraping, keeping only the document/page outline.
- **pauseAfter** waits, after the page is processed, for the passed number of seconds.
- **pauseBefore** waits, before the page is processed, for the passed number of seconds.
- **processPatterns** *(Expert & Enterprise)* interprets generation patterns and adds generated strings to the queue.
- **queueMax** *(Expert & Enterprise)* sets the maximum number of URLs to be added to the queue.
- **readFromQueries** *(Expert & Enterprise)* Reads the next active string from the passed query directory and stores its value in the passed variable, then unchecks the line in the query directory.
- **reapply** *(Expert & Enterprise)* reapplies the scraper without moving to the next page. (now accepts parameters for the number of reapplies and the delay between them.)
- **removeScripts** *(Expert & Enterprise)* cleans source code of the page before scraper is applied.
- **removeTags** *(Expert & Enterprise)* cleans source before scraper is applied, leaving only simple text.
- **rename** *(Expert & Enterprise)* renames all files matching a pattern in a disk folder using the replacement string.
- **repeat** adds the extracted field content to each extracted record.
- **replace** Pre-processing replacement
- **replaceInField** *(Expert & Enterprise)* replaces value in a given field.

- **replaceUsingQueries** *(Expert & Enterprise)* replaces matches of lists of patterns with lists of replacement strings.
- **resetAll, resetVisited, resetQueue** *(Expert & Enterprise)*reset lists of visited URLs, URLs to visit.
- **resetAllIf, resetVisitedIf, resetQueueIf** *(Expert & Enterprise)* conditional reset lists of visited URLs, URLs to visit
- **resetPrefOnStop** *(Expert & Enterprise)* Reset the passed preference to its default value at the end of the scrape process.
- **resetQueueIf** *(Expert & Enterprise)* resets queue of URLs to visit if scraper line matches.
- **resetQueueIfNot** *(Expert & Enterprise)* resets queue of URLs to visit if scraper line doesn't match.
- **resetVisitedIf** *(Expert & Enterprise)* resets list of visited URLs if scraper line matches.
- **resetVisitedIfNot** *(Expert & Enterprise)* resets list of visited URLs if scraper line doesn't match.
- **resetVisitedJSLinksIf** *(Expert & Enterprise)* resets list of visited Javascript links if scraper line matches.
- **restartEvery** *(Expert & Enterprise)* Sets 'auto-explore on startup' flag to true and restarts the application, every n pages or seconds.
- **save** *(Expert & Enterprise)* Saves the string extracted by the scraper line to a separate text file.
- **saveQueueAsQueries** *(Expert & Enterprise)* Sends the current queue of URLs left to visit to a query directory.
- **scope** *(Expert & Enterprise)* limits Fast mode explorations to a domain and/or a depth.
- **scrapeIf** conditions data extraction to a match in the source code.
- **scrapeIfIn** *(Expert & Enterprise)* proceed with extraction if URL is in query directory.
- **scrapeIfNot** conditions data extraction to no match in the source code.
- **scrapeIfNotIn** *(Expert & Enterprise)* proceed with extraction if URL is not in query directory.
- **screenshot** *(Expert & Enterprise)* Saves a screenshot of the current page into a file using the passed file name.
- **scrollBy** *(Expert & Enterprise)* Scrolls the page loaded in the OutWit Hub browser by the passed number of pixels.
- **scrollToBottom** *(Expert & Enterprise)* scrolls (once) to the bottom of the currently loaded page.
- **scrollToEnd** scrolls down to the end of the page before scraping.
- **scrollToString** *(Expert & Enterprise)* scrolls down to a string in the page.
- **sendToQueries** *(Expert & Enterprise)* sends extracted URL to the 'queries' view.
- **setAnchorRow** stores the row number for later reference.
- **setValue** *(Expert & Enterprise)* Sets the value of the <select> or <input> HTML block matching the format column, to the value passed in the replace column.
- **setVariable** *(Expert & Enterprise)* declares and sets the value of variable before the extraction process.
- **showAlert** displays an alert with the data scraped by the line.
- **showMatches** (or **logMatches**) displays alert (or message in console) with strings matching patterns.
- **showNextPage** (or **logNextPage**) displays alert (or message in console) with next page URL.
- **showNextPageCandidates** (or **logNextPageCandidates**) displays alert (or message in console) with list of next page candidates.
- **showOriginalSource** (or **logOriginalSource**) displays alert (or message in console) with source code before alterations.
- **showQueue** (or **logQueue**) displays alert (or message in console) with the queue.
- **showRecordDelimiter** (or **logRecordDelimiter**) displays alert (or message in console) with record delimiter.
- **showResults** (or **logResults**) displays alert (or message in console) with grabbed data.
- **showScraper** (or **logScraper**) displays alert (or message in console) with scraper content.
- **showScraperErrors** (or **logScraperErrors**) displays an alert (or a message in the error console) if an error occurs.
- **showServerErrors** creates a separate column with error messages.
- **showSource** (or **logSource**) displays alert (or message in console) with source code.
- **showVariables** (or **logVariables**) displays alert (or message in console) with values of variables.
- **showVisited** (or **logVisited**) displays alert (or message in console) with list of visited URLs.
- **simulate** processes the scraper without actually applying it.

- **skipIfIn** *(Expert & Enterprise)* does not visit page if the URL is present in query directory.
- **splitField** *(Expert & Enterprise)* Splits the passed field as a post-process, using the values in the separator and labels columns. (Can allow consecutive splits.)
- **start** starts extraction for the part of the source code following the match of this scraper line.
- **stop** stops extracting after the match of this scraper line
- **suspend, suspendIf, suspendIfNot, suspendEvery** *(Expert & Enterprise)* suspends all operations.
- **switchTo** *(Expert & Enterprise)* Changes the current view to the value set in the replace column.
- **switchTo** *(Expert & Enterprise)* changes the current view.
- **uncheckItemInQuery** *(Expert & Enterprise)* Unchecks the 'OK' checkbox of the first line containing the string extracted by the scraper line in the passed query directory.
- **uncheckURLInQuery** *(Expert & Enterprise)* Unchecks the 'OK' checkbox of the first line containing the current URL in the passed query directory.
- **uniqueField** *(Expert & Enterprise)* Makes sure that no duplicate values are extracted for the specified field(s) during the same exploration. (An alternative to deduplication while scraping, in case volumes are too large to post-process it.)
- **unzip** *(Expert & Enterprise)* Unzips a file or the content of a directory.
- **variable** declares and sets the value of a variable.
- **zapGremlins** removes unwanted control or invisible characters and tries to correct badly encoded characters.

Directive Details:

Pre-Processing:

- **#abortAfter#** Aborts the current extraction after the current page, if the scraper line matches.
- **#abortAfterNPages#n#** *(Expert & Enterprise)* Aborts the current extraction after the nth explored page.
- **#abortAfterNResults#n#** *(Expert & Enterprise)* Aborts the current extraction when the scraped view contains n extracted records or more.
- **#abortIf#** and **#abortIfNot#** abort the scraping and interrupt the current automatic exploration if the scraper line matches (or doesn't match) a string within the page.
- **#addURLToSource#** *(Expert & Enterprise)* adds the URL at the beginning of the source code between <pageurl></pageurl> tags, so that you can scrape the URL content or use it for conditional extractions. *Note that you will not see the added URL in the source code of the page displayed in the scraper editor or in the source view because the addition is only done when the scraper is applied. If you want to verify that it is there, you can use the #showSource# directive.*
- **#allFrames#** *(Expert & Enterprise)* Scrapes the concatenation of all frames in the current page.
- **#allowCrossDomain#** *(Expert & Enterprise)* Removes javascript same-origin policy restrictions and reloads the page. (Restrictions will be restored at end of process or at restart.)
- **#autoCatch#** *(Expert & Enterprise)* autoCatch:[true/false] - Forces the position of the Empty/Catch option of the scraped view to On Demand or Auto. (Empty On Demand and Auto-Catch will not be activated together.)
- **#autoEmpty#** & **#emptyOnDemand#** *(Expert & Enterprise)* directives set the value of the 'Empty' checkbox in the scraped view from within a scraper.
- **#caseSensitive#** makes the whole scraper case sensitive. Note that as every regular expressions and literals of a scraper are combined into a single regular expression at application time, it is therefore not possible to define case sensitivity line by line or field by field. The whole scraper must be conceived with this in mind.
- **#catchEvery#n# (or #catchAndDeleteEvery#n#)** *(Expert & Enterprise)* Sends, every n collected rows, the data stored in the scraped view (or in any other view specified in the Format column) to the Catch, then empties the datasheet if Delete is requested.
- **#checkIf#** and **#checkIfNot#** If the scraper line matches at least one string in the page, or does not match anything, or in any case, without condition (**#check#**) the content of the 'replace' field will alter the OK column of your scraper. A string of 0s and 1s in the replace field will set the OK checkboxes of the scraper in the same order. *Note that the right-click menu on the replace field of a #check directive line will allow you to copy the values from the OK column to the cell or copy the cell string to the OK column.*

Example:
You want to turn off line 5 of your scraper if the page doesn't contain "breaking news":
Description:
#checkIfNot#
Format:
breaking news
Replace:
11110111

- **#checkIfURL#** and **#checkIfNotURL#** allow you to include URL-based conditions in the scraper: If the scraper line matches (or does not match) the current URL, the content of the 'replace' field will alter the OK column of your scraper. A string of 0s and 1s in the replace field will set the OK checkboxes of the scraper in the same order. *Note that the right-click menu on the replace field of a #check directive line will allow you to copy the values from the OK column to the cell or copy the cell string to the OK column.*
- **#cleanHTML#** normalizes the HTML tags before the scrape, placing all attributes in alphabetical order. This can prove useful in some occasions when a page was typed manually (by a person lacking rigor), instead of generated automatically.
- **#clearAllHistory#** *(Expert & Enterprise)* Clears all history, forms, cookies, cache from the browser memory.
- **#clearBrowsingHistory#** *(Expert & Enterprise)* Clears browsing history.
- **#clearCookie#** *(Expert & Enterprise)* Clears cookies stored for the current URL.
- **#clearCookieEvery#n#** *(Expert & Enterprise)* Clears cookies for current URL every n pages.
- **#clearCookieIf#** *(Expert & Enterprise)* Clears cookies stored for the current URL if the scraper line matches.
- **#clearCookiesEvery#n#** *(Expert & Enterprise)* Clears all cookies every n pages.
- **#clearCookiesIf#** *(Expert & Enterprise)* Clears all cookies if the scraper line matches.
- **#clearCookiesIfNot#** *(Expert & Enterprise)* Clears all cookies if the scraper line does not match.
- **#clearForms#** *(Expert & Enterprise)* Clears form data from the browser memory.
- **#coalesceOnStop#criterionColumnName#** *(Expert & Enterprise)* Once the current automatic exploration is achieved or interrupted, data rows sharing the same value in the passed field name are coalesced keeping the first value found in each column.
- **#concatSeparator#separator#** allows you to set the character or string to be used as a delimiter in contactenation functions like #CONCAT#, #DISTINCT#, etc.
- **#deduplicate#** *(Expert & Enterprise)* changes the Deduplicate checkbox of the scraped view from within a scraper.
- **#deduplicateOnStop#criterionColumnName#** *(Expert & Enterprise)* removes duplicates from the extracted data (row by row) in the datasheet, only once the current automatic exploration is achieved or interrupted. (This prevents the deduplication from slowing down the whole extraction process.)
- **#decodeEntities#** Decodes HTML entities (like &amp; or &gt;) to their plain text equivalent in the whole source code before the scrape.
- **#decodeJSCharcodes#** Converts hexadecimal and decimal codes (\uXXXX and \xXX) into characters.
- **#deleteCookies#**, **#deleteCookiesIf#**, **#deleteCookiesIfNot#** *(Expert & Enterprise)* This directive will instruct the program to delete ALL cookies (respectively: in all cases, when the scraper line matches, when the scraper line doesn't match). Note that it will not only delete the cookies for the current page or domain, but all cookies for the application. (This means that you may for instance loose current session connections on other websites when using this directive.)
- **#downloadReferer#** *(Expert & Enterprise)* Sets the default referer for the next URL(s) downloaded in the current exploration.
- **#emptyOnDemand#** Forces the position of the Empty option of the scraped view to On Demand or Auto. If On Demand is set Auto-Catch is deactivated.
- **#emptyDirectory#queryDirectoryName#** *(Expert & Enterprise)* Empties the first directory of the queries view matching the passed directory [queryDirectoryName]. (Attention: no warning dialog)
- **#excludeFromQueueIf#** *(Expert & Enterprise)* Allows to exclude urls that match a string or a regular expression.

- **#excludeFromQueueIfNot#** *(Expert & Enterprise)* Allows to exclude urls that do not match a string or a regular expression.
- **#exportEvery# or #exportAndDeleteEvery#n#** *(Expert & Enterprise)* Exports, every n collected rows, the data stored in the scraped view (or in any other view specified in the Format column) to the file (defined by a filename, a renaming pattern or a path in the form of file://...) specified in the Replace column, Then empties the datasheet if requested.
  Note: If used in the Enterprise edition, this directive can define an SQLite database as the destination (using a filename with the .sqlite extension), which is the most efficient way to manage very large volumes of data.
- **#fieldGroup#** *(Expert & Enterprise)* Makes sure that the fields indexes in a same group are incremented together even if some of the fields are empty.
- **#getLocalIP#** *(Expert & Enterprise)* Gets public IP address of the machine from the network.
- **#getJSON#** *(Expert & Enterprise)* If this directive is used, the rest of the scraper will be ignored and the program will try to interpret JSON blocks found in the page. Note that converting complex recursive JSON objects into a two-dimensional table is only possible to a certain point. Do not expect to grab the data from complex objects this way. This function can however be very useful in simple cases.
- **#ignoreErrors#** When this directive is used, cells where a function returned an error will be empty instead of containing an ##Error message.
- **#indentedText#** alters the source code before scraping, reorganizing the document/page layout into an outline with indented text.
- **#insertIfNot#myFieldName#** If the scraper line does not match anything in the page, the content of the 'replace' field will be added once to each row scraped for this page. It is the only way to insert information to your extracted data when the page does *not* contain something.
- **#insertIf#myFieldName#** The data extracted by this scraper line will be added once to each record scraped in this page, if the scraper line matches one or more strings in the page. *It is mostly here as the corollary of the previous directive, but it is a good way to get rid of duplicate columns in certain cases.*
- **#keepOrder#** has the same effect as checking the 'keep order' checkbox in the scraped view or in a macro, i.e. ensuring that the columns will appear in the result datasheet in the same order as the scraper lines. Setting it directly in the scraper allows you to make sure to always have this behavior with this scraper. *Note that this directive is without effect on fields that are split using a separator.*
- **#keepForms#** Instructs the program not to remove forms when cleaning the extracted text;.
- **#limit#** *(Expert & Enterprise)* Only keeps the n first rows of extracted data.
- **#maxColumn#** *(Expert & Enterprise)* Sets the maximum number of columns in the extracted datasheet. Additional fields will be ignored.
- **#maxIndex#fieldName#** *(Expert & Enterprise)* Sets the maximum number of columns with the passed field name. Additional such columns will be ignored.
- **#minIndex#fieldName#** *(Expert & Enterprise)* Sets the minimum number of columns with the same name. Missing columns will be created empty.
- **#nextPageMax#** *(Expert & Enterprise)* Sets the maximum number of pages to be explored within the automatic Browsing of a series of pages.
- **#nextPageReferer#** *(Expert & Enterprise)* Sets the referer for the next page query that will be sent to the server.
- **#oneRow#** All data extracted in this page will be presented as a single row in the datasheet.
- **#originalData#** Does not clean the scraper result cells. Leaves all html tags and special characters.
- **#originalHTML#** Scrapes the original HTML source without any alteration, preparation or special character decoding.
- **#outline#** alters the source code before scraping, keeping only the document/page outline.
- **#pauseBefore#** instructs the scraper to wait, before the page is processed, for the number of seconds set in the Replace field.
- **#pauseAfter#** instructs the scraper to wait, after the page is processed, for the number of seconds set in the Replace field.
- **#processPatterns#** *(Expert & Enterprise)* instructs the scraper to check if URLs passed to the #addToQueue# directives are generation patterns. If they are, the patterns will be interpreted and all generated strings will be added to the queue.
- **#queueMax#** *(Expert & Enterprise)* Sets the maximum number of URLs to be added to the queue.

- **#readFromQueries#directoryName#** or **#readAndUncheckFromQueries#directoryName#** (*Expert & Enterprise*) Reads the next active string from the passed query directory and stores its value in the variable passed in the replace column (#myVariable#), then unchecks the line in the query directory.
- **#removeScripts#** (*Expert & Enterprise*) Cleans the source code of the page before the scraper is applied, by removing scripts and comments.
- **#removeTags#** (*Expert & Enterprise*) Cleans the source before the scraper is applied, removing html tags and unwanted characters, and leaving only simple text.
- **#rename#** (*Expert & Enterprise*) Renames all files matching the passed pattern in the passed directory using the replacement string. If no directory is provided, (...)/downloads/outwit/unzipped is used.
- **#replaceUsingQueries#myDirectory#** (*Expert & Enterprise*) Replaces matches of the string (or regExp pattern) in the 'query string' column of the directory with the content of the 'notes' column. (If the Replace column of the scraper line contains #RELOAD#, the page will be reloaded displaying the replacements.
- **#replace#** Pre-processing replacement: the string (or regular expression) entered in the 'format' field will be replaced by the content of the 'Replace' field throughout the whole source code of the page, before the scraper is applied.

  Example: The page you wish to scrape contains both "USD" and US$. You wish to normalize it before scraping:
  Description:
  #replace#
  Format:
  US$
  Replace:
  USD

- **#resetAll#, #resetVisited#, #resetQueue#** (*Expert & Enterprise*) reset the list(s) of visited URLs and/or URLs to visit (the queue).
- **#resetAllIf#, #resetVisitedIf#, #resetQueueIf#** (*Expert & Enterprise*) reset the list(s) of visited URLs and/or URLs to visit (the queue), when the scraper line matches a string in the page.
- **#resetQueueIf#** (*Expert & Enterprise*) Resets queue of URLs to visit if scraper line matches.
- **#resetQueueIfNot#** (*Expert & Enterprise*) Resets queue of URLs to visit if scraper line doesn't match.
- **#resetVisitedIf#** (*Expert & Enterprise*) Resets list of visited URLs if scraper line matches.
- **#resetVisitedIfNot#** (*Expert & Enterprise*) Resets list of visited URLs if scraper line doesn't match.
- **#resetVisitedJSLinksIf#** (*Expert & Enterprise*) Resets list of visited Javascript links if scraper line matches.
- **#restartEvery#** (*Expert & Enterprise*) Sets 'auto-explore on startup' flag to true and restarts the application, every n pages or seconds.
- **#scope#** (*Expert & Enterprise*) limits Fast mode explorations to a domain and/or a depth defined in the Replace column. Values of the desired exploration scope parameter can be: within_0, within_1, within_2, within_all, outside_0, outside_1, outside_2.
- **#scrapeIf#** Data will only be extracted from the page if this scraper line matches something in the page source code.
- **#scrapeIfNot#** Data will only be extracted from the page if this scraper line doesn't match anything.

  Example: You want to scrape only pages that contain "breaking news":
  Description:
  #scrapeIf#
  Format:
  breaking news

- **#scrapeIfIn#queryDirectoryName#** *(Expert & Enterprise)* Scrape the page if the URL is present in the passed query directory. (Doesn't work in fast scrape mode from a right-click on a list of URLs.).
- **#scrapeIfNotIn#queryDirectoryName#** or **#skipIfIn#queryDirectoryName#** *(Expert & Enterprise)* Proceed with the extraction if the URL is not found in the passed query directory.
- **#scrollBy#** *(Expert & Enterprise)* Scrolls the currently loaded page by the number of pixels set in the Replace column.
- **#scrollToBottom#** *(Expert & Enterprise)* Scrolls (once) to the bottom of the currently loaded page.
- **#scrollToEnd#** instructs the scraper to scroll down to the end of the page and wait for the number of seconds set in the replace field (usually for AJAX pages, in order to leave time for the page to be refreshed). You can also add a step parameter **#scrollToEnd#n#** to instruct the program to scroll down by steps of n pixels. (This is often useful in AJAX pages when the data is only loaded when the user scrolls to a certain part of the page.) You can finally use a css selector as a parameter (#scrollToEnd#cssSelector#), in order to address a specific HTML element and scroll down within this element. (This is very useful for recent AJAX interfaces.) *This function often requires some fine-tuning of the time parameters: the numer of seconds in the replace column of the scraper line, the three top sliders of Tools>Preferences>Timeouts... to define the max delays (they must allow for more time than the number of seconds set in the scraper), the AJAX settings in Tools>Preferences>Advanced... to define the behavior when new data is dynamically added to the page.*
- **#scrollToString#** *(Expert & Enterprise)* instructs the scraper to scroll down until the scraper line matches a string in the page (useful for AJAX pages, where new data is added when you scroll).
- **#setValue#** *(Expert & Enterprise)* Sets the value of the <select> or <input> HTML block matching the format column ("id=" or "class="), to the value passed in the replace column.
- **#setVariable#variableName#** *(Expert & Enterprise)* Declares and sets the value of the variable (#variableName#). Occurrences of #variableName# are replaced, before the extraction process, by the scraped value in all other lines of the scraper. Contrary to #variable#, #setVariable# sets the value of the variable before the source code is processed, so that even if this line matches at the end of the page only, the value will already be available when scraping the beginning of the page. Variables can only be used in the Description and Replace columns and only within the scope of one scraper execution. (They cannot be used to transfer information between two scrapers.).
- **#switchTo#** *(Expert & Enterprise)* Changes the current view to the value set in the replace column.
- **#uniqueField#** *(Expert & Enterprise)* Makes sure that no duplicate values are extracted for the specified field(s) during the same exploration. (An alternative to deduplication while scraping, in case volumes are too large to post-process it.)
- **#unzip#file:///.../myFile.zip#** *(Expert & Enterprise)* Unzips the file at the passed path (or all zip files in the first passed directory), into the directory passed in the replace column. If no directory is provided, (...)/downloads/outwit/unzipped is used.
- **#zapGremlins#** Removes unwanted control or invisible characters and tries to correct badly encoded characters.

Processing:

- **#addToQueue#** stores the data scraped by the line in a global variable. The queue can then be accessed with the #nextToVisit()# function. *(See below for more info.)*
  Note that in the Expert & Enterprise editions, when scraping the original source (white background), #addToQueue# is enough to instruct the program to visit grabbed URLs in Fast Scrape mode. In this case, #nextPage# and #nextToVisit()# are not needed.
- **#default#** or **#default#myFieldName#** *(Expert & Enterprise)* If this directive is used, the content of the 'Replace' field of the scraper line will be used as the default value for *myFieldName*. If *myFieldName* is not specified, the default will apply to all fields. This value will be added if the line doesn't match or is empty. (This is incidently a good way to solve the limitation problem of #keepOrder# with a separator.) *A scraper line using the #default# directive is not destined to directly grab data but to set the default value of a given field (or all fields) used in the scraper.*
- **#download#** or **#download#renamePattern#** *(Expert & Enterprise)* downloads the file at the URL grabbed by the scraper line (setting the renaming pattern if specified).
- **#extractContacts#n#** *(Expert & Enterprise)* Applies the contact extractor to the current URL loaded in the browser and returns all fields. The optional parameter sets the filter level.

Example:
Description:
  #download#
Marker Before:
  Product:<br /><img src="
Marker After:
  "

- **#exclude#myFieldName#** If this directive is used, the content of the 'Format' field of the scraper line will not be accepted as a value for *myFieldName*. If the line matches a string corresponding to the excluded value, the match will be ignored. *A scraper line using the #exclude# directive is not destined to grab data but to set an unwanted value for a field name used elsewhere in the scraper.*
- **#hideNodes#** *(Expert & Enterprise)* Makes the nodes matching the passed css selector invisible.
- **#newRecord#** Each time the pattern of this scraper line matches a string in the page source, a new record (new row) is created in the result datasheet. The pattern to match can be entered either in the 'marker before' field or in the 'format' field. *Note that if this directive is not used, the record separator is the first (top) record of your scraper. In cases where there is no clear field markers or if the field you want as a unique key for the record is not first, or if fields are not always populated, #newRecord# is very useful.*
- **#reapply#** *(Expert & Enterprise)* Reapplies the scraper without moving to the next page.
- **#repeat#myFieldName#** The matching or replacement value will be added in a column named myFieldName to all rows following the match of this scraper line. *A scraper line using the #repeat# directive is not destined to directly grab data. It instructs the program to add the field to each new extracted record; if no record is found by the scraper, the repeated field will not be inserted. (See the #insertIfNot# directive, if you want to return a value when a scraper doesn't match.)*

  Example: Say you have a page where the data to scrape is divided by continent between the following tags: <h3>Continent: XXXXX</h3>.
  You can set the scraper to add the continent in a column for every row by adding:
  Description:
    #repeat#Continent#
  Marker Before:
    <h3>Continent:
  Marker After:
    </h3>

  The repeat directive can be used to set a fixed value in a column by only entering a string in the Replace field:

  Example: For inputing data directly in your database without any touchup in the process you need to add the field "location" with a set value:
  Description:
    #repeat#Location#
  Replace:
    New Delhi

  Note: if a variable is entered in the Replace field, all its values will be concatenated in the repeated output.
- **#save#** *(Expert & Enterprise)* Saves the string extracted by the scraper line to a separate text file.
- **#saveQueueAsQueries#** *(Expert & Enterprise)* Sends the current queue of URLs left to visit to a query directory. A name can be added to the directive if you want to specify how the directory should be named or in order to replace an existing directory with the current

queue URLs: #saveQueueAsQueries#directoryName#. This directive can be extremely useful if you fear that a process may be interrupted. It will allow you to resume an automatic exploration where it stopped. Note, however, that the directive only stores URLs to explore, not URLs wihich have already been explored. If your scraper adds URLs to the queue, already visited URLs may be added again, so resuming an exploration this way may in some cases lead to re-visiting URLs that have been seen before the interruption.

- **#screenshot#** (*Expert & Enterprise*) Saves a screenshot of the current page into a file using the passed file name.
- **#sendToQueries#** (*Expert & Enterprise*) sends to the 'queries' view the URL extracted by the scraper line. A name can be added to the directive to specify how the directory should be called or to add the extracted URL to an existing directory: #sendToQueries#directoryName#.
- **#start#** switches scraping on. Data will start being extracted in the part of the source code following the match of this scraper line. *(Directives are not limited by #start# and #stop#. For instance, if the #scrapeIf# directive matches a string outside of the start/stop zones, it will still be executed.)*

Example: You only want to start scraping after a given title, say <h2>Synopsis:</h2>. You simply need to type the string in the Format field of your scraper line:
Description:
  #start#
Format:
  <h2>Synopsis:</h2>

- **#stop#** switches scraping off. Data extraction will stop after the match of this scraper line in the source code. (But the code analysis continues and scraping will start again if a #start# line matches.) *Note that if the #stop# line matches before a #start# line (or if there is no #start# line), a #start# directive is implied at the beginning. In other words, in order to be able to stop, the scraping needs to start. Directives are not limited by #start# and #stop#. For instance, if the #scrapeIf# directive matches a string outside of the start/stop zones, it will still be executed.*
- **#suspend#n#**, **#suspendIf#n#**, **#suspendIfNot#n#**, **#suspendEvery#n#** (*Expert & Enterprise*) This directive will instruct the program to suspend all operations (respectively: in all cases, when the scraper line matches, when the scraper line doesn't match, or every nth page). If you add a parameter to the first three functions, the program will wait for n seconds before resuming when the OK button is clicked. This is useful to give the user time to interract with the page, solve a captcha, etc.
- **#variable#myVariableName#** declares and sets the value of the variable (#myVariableName#). The occurrences of the variable are then replaced, at application time, by the scraped value in all other lines of the scraper. Variables can only be used in the Description and Replace columns and only within the scope of one scraper execution. They cannot be used to transfer information between two scrapers.

Example: Setting and using the variable 'trend'.

line 1:
Description:
  #variable#trend#
Marker Before:
  Dow Jones:
Marker After:
  <br />
Format:
  /[-+\d,.]+/

line 2:
Description:

```
  #showAlert#
Replace:
  #if(#trend#<0,Bear,Bull)#
```

- ***Anchor Functions:***(*The need for these functions is relatively rare, it will help you solve difficult cases when the data is presented in columns in the HTML page, using blocks with left or right 'float' tags.)* **#setAnchorRow#** stores the row number where this scraper line matches, so that data that will be found later in the page source code can be added to the result table as additional columns, starting at this row number. Thus, when the directive **#useAnchorRow#** is encountered --and if an anchor row has been previously set-- the following fields of data are added, starting at the anchor row until the **#useCurrentRow#** directive reverts to the normal behavior, adding a new row at the bottom of the result table each time a record separator is found.
  Post-Processing:
- **#alertOnStop#** Displays an alert at the end of the automatic exploration process (when no more next page link is found), with the list of visited URLs and discarded next page link candidates, if any.
- **#catchOnStop#** *(Expert & Enterprise)* Sends the data stored in the scraped view (or in any other view specified in the Format column) to the Catch at the end of the exploration.
- **#cleanData#** and **#originalData#** override the 'Clean Text' checkbox in the scraped view. When original data is set, the data is left as is (including HTML tags and entities), when clean data is used, HTML tags are removed from the scraped data.
- **#deduplicateWithinPage#** *(Expert & Enterprise)* Does a smart deduplication of the extracted data (row by row) for each scraped page, before sending the results to the datasheet. (This prevents the deduplication of tens of thousands of rows to slow down the whole process.).
- **#holdResults#**, **#holdResultsIf#** and **#holdResultsIfNot#** *(Expert & Enterprise)* are used to merge data scraped from different pages. They instruct the program to not return the extracted data in the present scrape but store it in memory until a #mergeResults# directive is used scraping another page. The conditional versions of the directive allow you to hold the set of results only if the scraper line matches or doesn't match a string in the page.
- **#insertIfNoResults#** Adds a line with the content of the Replace column if no data was extracted in the page.
- **#insertRow#** *(Expert & Enterprise)* Inserts a row with the value of the replacement column before (or after) every record of the final output (ignoring first and last).
- **#mergeResults#**, **#mergeResultsIf#** and **#mergeResultsIfNot#** *(Expert & Enterprise)* instruct the program to concatenate the previously stored data with the data scraped in this page and return the combined set of results. The conditional versions of the directive allow you to merge the results only if the scraper line matches or doesn't match a string in the page.
- **#nextPage#** allows you to tell OutWit Hub how to find the link to the next page to use in an automatic browse process. Use this when the Hub doesn't find the next page link automatically, or when you wish to manually set a specific course for the exploration.

  NOTE: As any feature in scrapers, the next page directive is only applied when the scraped view is active (which means that the view's bottom panel has non-default settings and the view name is in bold in the side panel).

Example: A typical next page scraper line.

```
Description:
  #nextPage#
Marker Before:
  <a href="
Marker After:
  ">Next page</a>
Format:
```

/[^"]+/
Replace:
#BASEURL#\0

- **#nextPage#x#** You can add a positive integer rating in the next page directive: if several nextPage directives are used, the first matching line of the highest rating will be chosen. Use #nextPage#0#, the lowest, for the default value. If #nextPage# is used without a rating parameter, it will be considered as the highest rated.

Example: You want to go to the link of the text "Next Page", if found, or go back to the previous page otherwise:

line 1:
Description:
#nextPage#0#
Replace:
#BACK#

line 2:
Description:
#nextPage#3#
Marker Before:
<a href="
Marker After:
">Next page</a>
Replace:
#BASEURL#\0

- **#normalizeToK#myFieldName#** and **#normalizeToUnits#myFieldName#** normalizes numerical value in the field *myFieldName*: converts it to decimal units (m, m^2, m^3, g...) or k units (km, km^2, kg...), removes thousand separators and uses the dot as a decimal separator. *A scraper line using a normalize directive is not destined to grab data but to instruct the program to normalize a field used elsewhere in the scraper.*
- **#replaceInField#aFieldName#** (*Expert & Enterprise*) replaces value (litteral of RegExp) in a given field at the end of the process.
- **#resetPrefOnStop#** (*Expert & Enterprise*) Resets the passed preference to its default value at the end of the scrape process.
- **#splitField#** (*Expert & Enterprise*) Splits the passed field as a post-process, using the values in the separator and labels columns. (Can allow consecutive splits.)
- **#uncheckURLInQuery#** (*Expert & Enterprise*) Unchecks the 'OK' checkbox of the first line containing the current URL in the passed query directory.
- **#uncheckItemInQuery#** (*Expert & Enterprise*) Unchecks the 'OK' checkbox of the first line containing the string extracted by the scraper line in the passed query directory.

- *Debug directives:*
  - **#showAlert#** displays an alert with the data scraped by the directive line. If only the 'Replace' field is filled, the alert will be shown at the end of the scraping.
  - **#showMatches#** (or **#logMatches#**) displays an alert (or a message in the error console) with all the strings that match the scraper patterns.
  - **#showNextPage#** (or **#logNextPage#**) displays alert (or message in console) with value of the selected next page URL.
  - **#showNextPageCandidates#** (or **#logNextPageCandidates#**) displays alert (or message in console) with list of possible next page URLs found.
  - **#showQueue#** (or **#logQueue#**) displays alert (or message in console) with content of the queue (the list of URLs to explore).
  - **#showRecordDelimiter#** (or **#logRecordDelimiter#**) displays alert (or message in console) with name of the field selected as the record delimiter for this scraper.

- **#showResults#** (or **#logResults#**) displays alert (or message in console) with data grabbed by the scraper.
- **#showScraper#** (or **#logScraper#**) displays alert (or message in console) with content of the scraper as interpreted by the program.
- **#showScraperErrors#** (or **#logScraperErrors#**) displays an alert (or a message in the error console) if an error occurs. (Most of the time alerts are not welcome as they would block the execution of automatic tasks.)
- **#showServerErrors#** creates a separate column in the result datasheet with error messages returned by the server.
- **#showSource#** (or **#logSource#**) displays alert (or message in console) with source code to which the scraper is applied (after replacements made by the #replace# directive).
- **#showOriginalSource#** (or **#logOriginalSource#**) displays alert (or message in console) with original source code that was sent to the scraper (before alterations).
- **#showVariables#** (or **#logVariables#**) displays alert (or message in console) with values of all variables.
- **#showVisited#** (or **#logVisited#**) displays alert (or message in console) with list of the URLs visited since the beginning of the browse process.
- **#simulate#** instructs the program to process the scraper without actually applying it. The interpretation is performed and some directives will still work, allowing you to display information for debug. This can be helpful if the scraper application fails --in particular in case of freezes during the application of scrapers with too complex or faulty regular expressions-- in order to seek the cause of the problem.

Time Variables (pro version)

The following variables can be used in the 'Replace' field to complement or replace the scraped content.

Use **#YEAR#**, **#MONTH#**, **#DAY#**, **#HOURS#**, **#MINUTES#**, **#SECONDS#**, **#MILLISECONDS#**, **#DATE#**, **#TIME#**, **#DATETIME#** in the 'Replace' field to insert the respective values in your replacement string.

Example:
You can add a collection time to the scraper using both a directive and a time variable:
Description:
#repeat#Collected On
Replace:
#DATETIME#

(*Expert & Enterprise*) **#DATE#mm/dd/yy#** (or another pattern between the last two sharp signs) can be added to specify the date format to be used.

Navigation Variables (pro version)

The following variables can be used in the 'Replace' field to complement or replace the scraped content.

- Use **#URL#**, **#BASEURL#**, **#DOMAIN#**, **#BACK#**, **#FORWARD#**... in the 'Replace' field to insert the respective values in your replacement string.
  - **#URL#**: current page URL *(http://www.example.com/whatever/page.htm)*
  - **#BASEURL#**: current page path *(http://www.example.com/whatever/)*

Example:
You grab a relative link (incomplete) from the source and want to absolutize it:
Description:
Link
Before:

```
  href="
After:
  "
Replace:
  #DOMAIN#\0
```

- **#DOMAIN#**: current domain *(http://www.example.com)*
- (*Expert & Enterprise*) **#DOMAIN-NAME#**: current domain name *(example)*
- (*Expert & Enterprise*) **#WITHIN-DOMAIN#**: is equivalent to #DOMAIN# but only yields a result if the link is within the current domain
- (*Expert & Enterprise*) **#OUTSIDE-DOMAIN#**: only yields a result if the link is not within the current domain *(http://blog.example.com is accepted when scraping http://www.example.com)*
- (*Expert & Enterprise*) **#OUTSIDE-DOMAIN-NAME#**: is equivalent to #OUTSIDE-DOMAIN# but only yields a result if the link is not within the current domain name *(http://blog.example.com is not accepted when scraping http://www.example.com)*
- **#BACK#**: previous page in history
- **#FORWARD#**: next page in history

Example:
You just want the source domain in a column 'Source':
Description:
repeat#Source
Replace:
Collected on #DOMAIN#

- (*Expert & Enterprise*) Use **#RELOAD#** for conditional reloading of the current page.
- Redirections:
  - **#REQUESTED-URL#** gives the URL that was queried or clicked on.
  - **#REDIRECTED-URL#** returns the URL the browser eventually landed on after a redirection, if any, and returns nothing if there was no redirection.
  - **#TARGET-URL#** returns the URL the browser eventually landed on after a redirection, if any, and returns the requested (current) URL if there was no redirection.
- (*Expert & Enterprise*) Click Generation:
  - **#CLICK-ID#id#** When used in the Replace column on a #nextPage# line, the program will simulate a click on the HTML node with the provided ID.
  - **#CLICK-CLASS#classname#** When used in the Replace column on a #nextPage# line, the program will simulate a click on the first HTML node with the provided Class name.
  - **#CLICK-CLASS-FIRST-NODE#classname#** When used in the Replace column on a #nextPage# line, the program will simulate a click on the first node of the provided Class.
  - **#CLICK-CLASS-LAST-NODE#classname#** When used in the Replace column on a #nextPage# line, the program will simulate a click on the last node of the provided Class.
  - **#CLICK-CLASS-NODEn#classname#** When used in the Replace column on a #nextPage# line, the program will simulate a click on the nth node of the provided Class.
  - **#CLICK-CLASS-FIRST-LINK#classname#** When used in the Replace column on a #nextPage# line, the program will simulate a click on the first hyperlink with the provided Class name.
  - **#CLICK-CLASS-LAST-LINK#classname#** When used in the Replace column on a #nextPage# line, the program will simulate a click on the last hyperlink with the provided Class name.
  - **#CLICK-CLASS-LINKn#classname#** When used in the Replace column on a #nextPage# line, the program will simulate a click on the nth hyperlink with the provided Class name.

    - **#CLICK-CLASS-ALL#classname#** When used in the Replace column on a #nextPage# line, the program will simulate a click on all nodes with the provided Class name.
    - **#CLICK-CLASS-NEXT-LINK#classname#** When used in the Replace column on a #nextPage# line, the program will successively simulate a click on all links with the provided Class name, and run a scrape on each resulting page.
    - **#CLICK-CLASS-NEXT-NODE#classname#** When used in the Replace column on a #nextPage# line, the program will successively simulate a click on all nodes with the provided Class name, and run a scrape on each resulting page.
    - **#CLICK-SELECTOR#cssSelector#** When used in the Replace column on a #nextPage# line, the program will simulate a click on the first hyperlink matching the provided css selector.
    - **#CLICK-SELECTOR-FIRST-NODE#cssSelector#** When used in the Replace column on a #nextPage# line, the program will simulate a click on the first node matching the provided css selector.
    - **#CLICK-SELECTOR-LAST-NODE#cssSelector#** When used in the Replace column on a #nextPage# line, the program will simulate a click on the last node matching the provided css selector.
    - **#CLICK-SELECTOR-NODEn#cssSelector#** When used in the Replace column on a #nextPage# line, the program will simulate a click on the nth node matching the provided css selector.
    - **#CLICK-SELECTOR-FIRST-LINK#cssSelector#** When used in the Replace column on a #nextPage# line, the program will simulate a click on the first hyperlink matching the provided css selector.
    - **#CLICK-SELECTOR-LAST-LINK#cssSelector#** When used in the Replace column on a #nextPage# line, the program will simulate a click on the last hyperlink matching the provided css selector.
    - **#CLICK-SELECTOR-LINKn#cssSelector#** When used in the Replace column on a #nextPage# line, the program will simulate a click on the nth hyperlink matching the provided css selector.
    - **#CLICK-SELECTOR-ALL#cssSelector#** When used in the Replace column on a #nextPage# line, the program will simulate a click on all nodes matching the provided css selector.
    - **#CLICK-SELECTOR-NEXT-NODE#cssSelector#** When used in the Replace column on a #nextPage# line, the program will successively simulate a click on all nodes matching the provided css selector, and run a scrape on each resulting page.
- (*Expert & Enterprise*) Sounds:
    - **#BEEP#**, **#TICK#**, **#CHIMES#**, **#WOOSH#** produce a sound when the scraper line matches a string in the page.
- (*Expert & Enterprise*) Host Info:
    - **#HOSTNAME#** returns the most probable name of the organization hosting the current Web page.
    - **#HOSTCOUNTRY#** returns the most probable country of the current Web page.
- **#ORDINAL#** returns the ordinal number of the page being scraped in an automatic exploration. *(Note that this is different from the Ordinal ID column in datasheets. The number returned by #ORDINAL# is the first group of digits that constitute the Ordinal ID.)*
- **#COOKIE#** returns the content of the cookie(s) that have been set in your browser by the current Website if any.

Data Cleaning

Several features allow you to decide how the data should be cleaned from the HTML tags. The **'Clean Text'** checkbox of the scraped view bottom panel is the most obvious. It allows to set the application to remove/keep the HTML tags in all scrapers. The **#cleanData#** directive in a scraper allows you to set the behaviour for this scraper. Finally, the Description field (first column) can be used to set the cleaning behavior for a single field: **<MyFieldName>** keeps all HTML tags, **`*MyFieldName*`** returns the scraped data with a streamlined HTML, only keeping formatting tags and **MyFieldName** returns text without any HTML or style.

Required Fields *(Expert & Enterprise)*

! And ? Suffixes in the description column allow you to specify which field(s) must not be empty, in order to return a result. ! means "required field" (the record will not be returned if this field is empty) and ? means "Ignored if the other fields are empty" (this field will not be returned if the other fields are empty). Example: **MyFieldName?**

Indexed Columns *(Expert & Enterprise)*

Alternatively to the #maxIndex# directive, the syntax **myFieldName<n** in the description column allows you to specify the maximum number of columns with the same name (subsequent matches for the field are ignored).
myFieldName= in the description column prevents duplicate values for this field in the same record.

Replacement functions *(Pro, Expert & Enterprise Editions)*

The following functions can be used in the 'Replace' field to alter the scraped content.

These are executed when the scraper line (markers and/or format) match a string in the source code.

> *NOTE: these functions are still subject to evolution. At this point they can only be used alone in the replace field. They can now be used in a variable declaration.*

- Put **#AVERAGE#**, **#SUM#**, **#MAX#, #MIN#, #CONCAT#, #HAPAX#, #UNIQUE#, #STRICTLY-UNIQUE#, #DISTINCT#, #STRICTLY-DISTINCT#, #FIRST# to #FIFTH#, #LAST#, #SHORTEST#** or **#LONGEST#** in the 'Replace' field to replace the scraped values by the corresponding total calculation*. (Note that totals cannot serve as record separator. They will only work if not located on the first line of a scraper.)*
    - **#AVERAGE#**: if scraped values are numerical, the result is replaced by the arithmetic mean of these values
    - **#SUM#**: if scraped values are numerical, the result is replaced by the sum of these values
    - **#MIN#**: if scraped values are numerical, the result is replaced by the minimum value, otherwise by the first in alphabetical order
    - **#MAX#**: if scraped values are numerical, the result is replaced by the maximum value, otherwise by the last in alphabetical order
    - **#CONCAT#**: all values are concatenated, using semicolons as separators
    - **#COUNT#**: the number of occurrences
    - **#HAPAX#**: if only one occurrence is found, it is returned, otherwise the field does not return anything
    - **#UNIQUE#**: if only one value is found (whatever the number of occurrences), the value is returned, otherwise the field does not return anything
    - **#STRICTLY-UNIQUE#**: (case sensitive) if only one value is found (whatever the number of occurrences), the value is returned, otherwise the field does not return anything
    - **#DISTINCT#**: all distinct values are concatenated, using semicolons as separators; duplicate values are ignored (even if in different cases)
    - **#STRICTLY-DISTINCT#**: (case sensitive) all distinct values are concatenated, using semicolons as separators; exact duplicates are ignored
    - **#DISTINCT-COUNT#**: creates two columns (fields). The first one with the COUNT, the second with the DISTINCT concatenation.
    - **#STRICTLY-DISTINCT-COUNT#**: creates two columns (fields). The first one with the COUNT, the second with the STRICTLY-DISTINCT concatenation.
    - **#FIRST#,#SECOND#,#THIRD#,#FOURTH#,#FIFTH#**: only the corresponding occurrence is returned
    - **#LAST#**: only the last occurrence is returned
    - **#SHORTEST#**: only the shortest matching occurrence is returned
    - **#LONGEST#**: only the longest matching occurrence is returned
    - **#MAXLENGTH#**: the result is replaced by the maximum length (number of characters) of matching strings
    - **#MINLENGTH#**: the result is replaced by the minimum length (number of characters) of matching strings

Totals and Concat functions are now allowed in variable declarations and in replacement functions or operations.

(Expert & Enterprise) adding -INREC to the above functions (**#MAX-INREC#**, **#FIRST-INREC#**, **#DISCTINCT-INREC#** etc.), will return a total or concatenation for each record. This means that the function will be calculated for each block of source code between two occurrences of the record separator (the first field of the scraper, in general).

- **Operations: #(term1 operator term2)#** Works with the following operators: **+** (addition of integers: 1+3=4; concatenation of strings: out+wit=outwit; incrementing characters: c+3=f), **-** (subtraction of integers: 5-2=3 or decrementing chars: e-3=b), ***** (multiplication), **/** (division), **^** (power), **<**, **>**, **=**, **==**, **!=**,... (comparison operators): a=A (case-insensitive comparison), a==a (case-sensitive comparison), a!=b (not equal, case insensitive), a!==b (not equal, case sensitive). The terms can be literals, variables or functions. When using equality operators on strings (=, !=, ==, !==), you can now use the wildcard % in the second term to replace any string. (ex. these three statements are true: headstart = Head% ; homeland == h%d ; lighthouse = %HOUSE).
- **Conditions:**
  #if(condition,valueIfTrue,valueIfFalse)# or
  #if(condition;valueIfTrue;valueIfFalse)# for conditional replacements. The separator used between the parameters (comma or semicolon) must not be present in the parameters themselves.
- *(Expert & Enterprise)* **Dates:**
  - **#formatDate(string)#** tries to convert the scraped string into date and formats it in the standard and easily useable format of yyyy-mm-dd hh.mm.ss
  - **#parseDate(string)#** tries to convert the scraped string into a number which can be compared to another, for a conditionnal extraction, for instance. "Past" can be translated in the replace column by:
    #parseDate(\0)#<#parseDate(#DATE#)#.
- *(Expert & Enterprise)* **Names:**

  The following features are statistics-based recognition functions. They are extremely useful to enhance large quantities of data but they cannot be 100% accurate. A good idea is to store the data into additional fields, and also extract the original full name field, so that you can compare the results.

  - **#firstName(string)#** tries to find the most likely first name in the passed full name string.
    If the scraper line extracts "Dr Peter de Witt, M.D. 1998", the replace expression #firstName(\0)# will return "Peter".
  - **#lastName(string)#** tries to find the most likely last name in the passed full name string.
    If the scraper line extracts "Dr Peter de Witt, M.D. 1998", the replace expression #lastName(\0)# will return "de Witt".
  - **#firstLastName(string)#** tries to find the most likely first and last name in the passed full name string.
    If the scraper line extracts "Dr Peter de Witt, M.D. 1998", the replace expression #firstLastName(\0)# will return "Peter de Witt".
  - **#gender(string)#** tries to find the most likely gender for the passed full name string.
    If the scraper line extracts "Dr Peter de Witt, M.D. 1998", the expression #gender(\0)# will return "M".
    If the gender cannot be determined, the function returns "-". (Gender is returned for recognized first names with distribution statistics higher than 65% or 70%.)
- *(Expert & Enterprise)* **Transformation:**
  - **#base64(url)#** can be used to convert a small image into a self-contained data element using the data: URI scheme. Inline images allow to embed images within a web page or a document, as immediate (offline) data that the browser can render without querying the Web.
  - **#encodeURL(string)#** converts some special caracters to their hexadecimal equivallent so that they can be used in URL parameters.
  - **#decodeURL(string)#** decodes URL encoded characters (like %20) to their plain text equivalent.
- **Lookup lists:**
  #lookUp(originalString,listOfValuesToFind,listOfReplacementValues)# or
  #lookUp(originalString;listOfValuesToFind;listOfReplacementValues)# or

#lookUp(originalString|listOfValuesToFind|listOfReplacementValues)# for replacing lists of values. The parameters listOfValuesToFind and listOfReplacementValues must include the same number of items, separated by commas or semicolons. The elements of the first list will be respectively replaced by those of the second. The separator used between the parameters must not be present in the parameters themselves. (originalString can typically be the result of the scraper line \0 or a variable #myVariable# etc.)

- **Replace function** ***(not to be confused with the replace directive)*** **#replace(originalString,stringToFind,replacementString)#** or **#replace(originalString;stringToFind;replacementString)#** or **#replace(originalString|stringToFind|replacementString)#** replaces the first occurrence of stringToFind by replacementString in originalString. (originalString can typically be the result of the scraper line \0 or a variable #myVariable# etc.)
- **URL alteration functions: #getParam(URL,parameterName)#** returns the value of a parameter in the passed URL and **#setParam(URL,parameterName,parameterValue)#**, to assign a new value to a parameter. When used in conjunction with #URL# in the #nextPage# directive line, this function allows you to easily set the value of the next page URL in many cases. (URL can typically be the result of the scraper line \0 or a variable #myURL#, or a replacement function #URL#, #REDIRECTED-URL# etc.)
- **Alert: #alert(Your Message)#** Displays an alert with the message passed as a parameter (and blocking the scraping process).

Example:
This scraper line will generate the next URL to explore, incrementing the parameter 'page' in the current URL.
Description:
#nextPage#
Replace:
#setParam(#URL#,page,#(#getParam(#URL#,page)#+1)#)#

Automatic Exploration and Hierarchical Scraping *(Pro, Expert & Enterprise Editions)*

It is now possible for a scraper to set the URL of the next page to explore in a browse process (see #nextPage# directive above). Together with this feature comes a replacement function which allows advanced users to develop powerful scraping agents:

#nextToVisit(#myURL#)#, in the 'Replace' field, instructs the Hub to give the variable #myURL# the next value which is not found in the list of visited URLs. If you set #variable#myURL# in a scraper line, and if this line matches say 10 strings within the source code of the page, this variable will contain an array of 10 values. The #nextToVisit# directive will give #myURL# the value of the first URL which hasn't been explored in the current Browse process. This means that, used in conjunction with #nextPage# and #BACK# you can create complex scraping workflows. You can, in particular, create multi-level scraping processes.

#addToQueue# and **#nextToVisit()#**: This follows exactly the same principle, but without declaring a variable. It is simpler to use but it offers a little less control as it only allows you to have a single stack of URLs to explore. Contrary to variables, the queue can be accessed by any scraper during the process of an exploration. You can put URLs in the queue with one scraper and refer to it with another.

Example 1: Two-level scraping using **#addToQueue#** and **#nextToVisit()#**
Say you have a page named 'Widget List' with a list of URLs leading to the 'Widget Detail' pages where the interesting information is. You just need to create two scrapers:

Scraper #1:
Apply if URL contains:
widget-list

line 1:
Description:
#addToQueue#
Marker Before:
<a href="
Marker After:
">See Widget Description</a>
Replace:
#BASEURL#\0

Line 2:
Description:
#nextPage#
Replace:
#nextToVisit()#

Scraper #2:
Apply if URL contains:
widget-detail

line 1:
Description:
#nextPage#
Replace:
#BACK#

line 2...:
... scrape the data here.

Example 2: Two-level scraping using a variable **#nextToVisit(#extractedURLs#)#**
Same scenario, but this time, using a variable (for instance because you wish to keep two different kinds of URLs in separate piles):

Scraper #1:
Apply if URL contains:
widget-list

line 1:
Description:
#variable#extractedURLs#
Marker Before:
<a href="
Marker After:
">See Widget Description</a>
Replace:
#BASEURL#\0

Line 2:
Description:
#nextPage#
Replace:
#nextToVisit(#extractedURLs#)#

Scraper #2:
Apply if URL contains:

widget-detail

line 1:
Description:
#nextPage#
Replace:
#BACK#

line 2...:
... scrape the data here.

Note: This may look confusing, but it's not all that bad, once you have gotten the principle. The idea is that you often have a list L1 that links to another list L2 (n times), which in turn links to the pages P where you want to scrape your data.

Think of it from the end:

- *You have to make a page scraper (#2 in the example above) for the data in P with #nextPage# set to #BACK# (It's the "leaf" at the end of the branch, so the program will backtrack once the page is scraped.)*
- *You also have to make one or several list scrapers where you will extract the links from L1, L2... into a variable like #extractedURLs#.*
- *In the list scraper, you also need to set #nextPage#1# (higher priority) to #nextToVisit(#extractedURLs#)# to explore all the pages one after the other,*
- *and, finally -still in the list scraper- set #nextPage#0# (default value) to #BACK#, to backtrack to the higher level, once all #extractedURLs# of the level have been visited.*

One of the tricky things is to make sure that each scraper will apply to the right kind of page using the "URL contains" field. This may require a regular expression.

In the Expert & Enterprise editions, when scraping the original source (white background), #addToQueue# is enough to instruct the program to visit grabbed URLs in Fast Scrape mode. In this case, #nextPage# and #nextToVisit()# are not needed.

Applying a Scraper to a Page *(or Series of Pages)*

If you simply want to apply the best matching scraper to the current URL (the page loaded in the *page* view), just go to the scraped view. If you want to apply it to a series of pages or to the content of a site, you can set the scraped view's bottom panel as you want, (uncheck 'Empty' to keep the results in the scraped view OR check 'Catch selection' to move them to the catch) and use the Browse or Dig commands to explore the pages you want.

If you need to apply a scraper to a whole list of URLs, another direct way is to select the rows containing the links you want to scrape (in any view: usually 'queries' or 'the Catch', but it can also be 'links', 'tables', 'lists', 'guess'...), then right-click (ctrl-click on Macintosh) on one of the URLs to scrape (they should all be in the same column) and, in the contextual menu, select '**Auto-Explore**' > '**Fast Scrape**'. For each of the selected URLs the resulting data will be added in the datasheet of the *Scraped* view. (A throbber beside the view name shows that the process is running.) Of course, the scrape can also be done automatically with a macro.

Browsing & Scraping vs. Fast Scraping

> *Note that these two methods are very different: when a scraper is applied to the* ***Dynamic source code*** *during a browse, the data comes from the page loaded in the Hub's browser (light yellow background in the 'source' view), whereas when using the 'Fast Scrape' function on Selected URLs, the program runs an XML HTTP Request for each URL, but doesn't render the pages returned by the server. Thus, OutWit doesn't have to load all elements (images, sounds, dynamically added data, etc.) and can perform the scrape directly and much faster. The source used in this case is the* ***Original source code*** *transmitted from the server (white background in the 'source' view). In the majority of cases (in static HTML pages), the extracted results are the same, the 'Fast Scraping Mode' being simply much faster. In other cases, however, the pace can be too high for the server, the results can be different or the fast scraping mode can even completely fail: the reason is that in the normal (Browse) mode, events can happen that dynamically alter a page (mostly due to the execution of javascript scripts in AJAX pages). These dynamic changes do not occur in the Fast Scrape mode, as scripts are not executed. This means that dynamically added information, javascript redirections, page reloads... do not happen in fast scraping mode. If you notice this kind of behavior, if info is missing in the Original source (white background), the only way is to accept the slower method and browse through the URLs, doing the scraping page after page.*

Important: The Expert & Enterprise editions have the ability to populate the queue of URLs to visit during a Fast Scrape. It means that for non-AJAX sites (if what you want to scrape is in the Original source code), you can scrape the links to explore and visit them in the same process, as you would in Browse mode. You can also Dig a whole Website in Fast Mode with Expert and Enterprise, at a speed that can often be 10 or 20 times what the Pro edition could achieve.

Temporization

You can set the exploration speed in the *Time Settings* tab of the *Preferences* panel (Tools menu). By default, the temporization between pages is set to 4 seconds in Browse mode but it can ba as low as .1 second in Fast Mode. Set the speed as you wish, but make sure to respect the sites' terms of use and that you are not overusing the servers.

If you want to mimic manual exploration, you can set different minimum and maximum temporization times and the program will wait for a randomly selected duration between these two values before loading the next URL.

Use of Regular Expressions

Regular Expressions are a powerful syntax, used to search specific patterns in text content. They can be used in several places of OutWit Hub:

1. In the bottom panel of each widget (images, links, contacts...) the **Select If Contains** text box allows you to select items of the list above it that contain the typed string. By starting and ending the string with the character **/** you can use Regular Expressions in these text boxes.
2. In the Scraper Editor located in the 'Scrapers" widget, **Marker Before** and **Marker After** can be either a literal string or a Regular Expression. **Format** is always interpreted as a Regular Expression.
3. Lastly, the 'URL to Scrape' attributed to a scraper can also be a regular expression. In this case, the scraper can be applied to any URL matching the pattern.

To use regular expressions, write your string between slashes: **/myRegExp/**. The pattern will be displayed in green when the syntax is correct, in red otherwise.

IMPORTANT NOTES:
The 'Format' field of the Scraper Editor is always interpreted as a regural expression, even if not marked with slashes.
For technical reasons, all regular expressions used in scrapers are interpreted as case insensitive patterns by default. [A-Z], [A-Za-z] and [a-z] have the same result. This can be changed using the #caseSensitive# directive.

Here is what you should know if you are using regular expressions:

- Ultra Quick Start Guide
- Quick Start Guide
- More

Regular Expressions Ultra Quick Start

Regular expression patterns are strings to match in a text, surrounded with **/** (slashes) and including a series of reserved characters used as wildcards (i.e. representing ranges of characters or remarkable features).

The three most useful patterns:

- use the pattern **\s*** to match a succession of zero or more space characters, tabs, returns, etc.
- use the pattern **[^<]+** to match a succession of one or more characters until the next **<**
- use the pattern **[a-z]+** to match a succession of one or more letters

The two mistakes you are most likely to make, once you have learned more about RegExps:

- The character **.** (dot) doesn't mean 'any character', but 'any character, except return characters' (returns, line feeds, form feeds etc.), so do not use **.*** to say 'anything'. Instead, you should use **[\s\S]***, for instance, which means any succession of non-space characters or space characters.
- Among the characters that need to be escaped in a RegExp pattern is the very common **/** (slash). If you forget to escape these, the regular expression will not work. You need to escape it like this: **\/** (backslash followed by slash).

Example:
The pattern `/<span[^>]+>\s*Phone\s*:/` will match any `<span...>` tag followed by Phone, followed by the colon character (:)
whatever the number of spaces, tabs or returns between these elements.

To learn some more about *Regular Expressions*, you can go to our RegExp Quick Start Guide

Regular Expressions Quick Start

Marking a Regular Expression: /myRegExp/

Most of the time, simple strings will be enough as markers or selection criteria. Such a literal string must be typed and will be searched *as is* in the data. Therefore, if you want to use a regular expression instead, you must mark it, so that the program can identify it as such. This is done by adding a **/** before and after the reg exp pattern.

Escaping Special Characters

Characters that are used in the regular expressions syntax, like .$*+-^\(){}[]/ should be 'escaped' when used literally in a regular expression (i.e. when used as the character itself, not as part of the reg. exp. syntax). Escaping means placing a backslash character \ before that special character to have it be treated literally. To search for a backslash character, for instance, double it \\ so that its first occurrence will escape the second.

Most common "special" characters in regular expressions

Wildcard

. (dot): any character except a line break (or carriage return)

Character Classes (Ranges of Characters)

In a character class, a caret character **^** excludes all characters specified by a character class, if placed immediately after the opening bracket [^...].

[abc] list: any of the character a, b, c

[^abc] exclusion list: any character except a, b, c

[a-z] range: any character from a to z

[^aeiou] any character which is not a vowel

[a-zA-Z0-9] any character from a-z, A-Z, or 0-9

[^0-9aeiou] any character that is neither a digit nor a vowel

Escaped matching characters

\r line break (carriage return)
\n Unix line break (line feed)
\t tab
\f page break (form feed)
\\ backslash

\s any space character (space, tab, return, line feed, form feed)
\S any non-space character (any character not matched by \s)
\w any word character (a-z, A-Z, 0-9, _, and certain 8-bit characters)
\W any non-word character (all characters not included by \w, incl. returns)
\d any digit (0-9)
\D any non-digit character (including carriage return)
\b any word boundary (position between a \w character and a \W character)
\B any position that is not a word boundary

Alternation

| (pipe): Separates two expressions and matches either

Position

^: (when not in a character class) beginning of string
$: end of string

Quantifiers

x*: zero or more x
x+: one or more x
x?: zero or one x
x{COUNT}: exactly COUNT x, where COUNT is an integer
x{MIN,}: at least MIN x, where MIN is an integer
x{MIN, MAX}: at least MIN x, but no more than MAX

> **Note:**
> **+** and ***** are 'greedy': they match the longest string possible. If you do not want this "longest match" behavior, you can use non-greedy quantifiers, by adding a **?**.
>
> ***?**: zero or more (non-greedy)
> **+?**: one or more (non-greedy)
> **??**: zero or one (non-greedy)
>
> For example, Instead of:
> */<SCRIPT>.*<\/SCRIPT>/*
> Use the non-greedy quantifier:
> */<SCRIPT>.*?<\/SCRIPT>/*

Using Parentheses for Grouping

By placing a section of a regular expression between parentheses (round brackets), you can group this section. This allows you to apply an operator, e.g. a repetition operator like **?+***, to the entire group or to use alternation in the section, using one or several pipes: **It's a (big|large|tall|wide) house**.

Parentheses also create a "backreference" list that you can use in the replacement pattern. This means that, *unless you use non-capturing parentheses xxx(?:yyy)zzz*, you can address a section between parentheses by typing **\1 \2 \3**... in the replace field, the number indicating which captured group you wish to insert in the replacement string.

> Format:**It's a (big|small) (house|apartment) in (London).**
> Replace:**City:\3,Size:\1,Type:\2**

More on The Regular Expressions Syntax

*The following is to be used in the **Select If Contains** or **Advanced Dig** filter boxes.(It is not applicable and will be ignored in **Scrapers**.)*

Position wildcards

^ beginning of a line (if not in a character class)
$ end of line (if not in a character class)

(You can combine **^** and **$** within a pattern to force a match to constitute an entire line. Example: **^Home$**)

Text Case

You can place the **i** flag at the end of your regular expression pattern to indicate that you want the matching to be case insensitive. **/myPattern/i** will match 'MyPattern' 'mypattern', etc.

The *Macros* View *(Pro, Expert & Enterprise Editions)*

The macros view gives access to the macro manager and editor.

Macros are stored configurations that will allow you to easily automate complex extraction tasks in a single command. The different extractions processes can be defined in the *Macro Editor*.

When executed, a macro does the following:

- saves the current settings of the application,
- sets all desired view settings for the execution of the macro,
- loads the start URL,
- executes all active extractions, exports and downloads, while performing the selected exploration (browse, dig, slideshow, fast scraping),
- exports the catch data to a folder of your hard disk or to an FTP server if specified,
- restores the original configuration.

All the macros of your profile are listed in the manager with their Automator ID (AID), which can be used to run the macro from the command line using the parameter: -macro xxx.

The *Macro Editor* *(Pro, Expert & Enterprise Editions)*

The macro editor allows you to create and edit macros.

In the top left corner of the Macro Editor Panel appears the **name** of the macro being edited, on a popup menu, which allows you to switch to any of the macros stored in your profile.

Start Page Leave this field blank if you wish the macro to be applied to the currently loaded page. If you wish to specify a starting page for the process, enter the full URL of the first page to which your macro will be applied. If OutWit finds a possible next page link and if you have checked *Browse*, *Dig* or *Slideshow* in the *Navigation Zone*, then the macro will be applied to a whole series of pages. Start Page can also contain a column of the Catch (in the form "**catch/theColumnName**" --or "**catchData/theColumnName**" if you want the other columns of your catch included in the final extraction), a Query Directory (in the form "**queries/theDirectoryName**" --or "**queriesData/theDirectoryName**" to keep the Notes column) or a Query Generation Matrix. In these cases, the macro will be applied to all URLs found in the Catch column, in a directory of queries or to all URLs generated by the matrix.

The following series of buttons gives you access to the general management functions:

- **Save** Saves the current macro to your profile
- **New** Opens a new blank macro
- **Reset** Empties the macro editor
- **Revert** Restores the last saved version of the macro
- **Delete** Deletes the current macro from your profile
- **Import** Loads a previously exported macro
- **Export** Saves the macro as an XML file to your hard disk
- **Get from Views** Copies the current configuration of the bottom panels of all views to the extractors panel of the macro editor
- **Send to Views** Copies the current configuration of all extractors in the macro editor to the bottom panel of their respective views
- **Properties** Displays the Macro Properties Dialog where information about the current macro can be found and edited (name, author, comments, etc.)
- **Close** Closes the Macro Editor and displays the Automator Manager.

Below the previous controls, the *Macro Editor* view is divided in several zones:

- Macro as URL (MAU)
- Navigation
- Catch
- Destination files
- Extractor settings

Macro as URL (MAU)

The *Macro as URL* is the representation of all the non-default settings as a single string that can be pasted in OutWit Hub's address bar for immediate execution, pasted in an email to be shared with other OutWit users etc. The MAU is updated in real time when the characteristics of the macro are changed in the macro editor. Reversely, editing the MAU will alter the definition of the macro in the user interface. *An invalid syntax will cause the MAU to be displayed in red.*

Note: You can execute a macro by simply pasting the MAU in the Hub's address bar and hitting return.

'Navigation' Zone

- **Browse** When checked, the macro will look for a next page link after the execution of all the active extractors in the current page, and so on, until no following page is found. The number of pages to explore this way can be limited in the Browse popup menu.
- **Dig** When checked, the active extractors will be applied to all links found in the start page (or of all browsed pages, if Browse is checked). The domain and depth of the links to explore can be defined in the Dig popup menu. The advanced settings dialog allows you to set a criterion to visit only certain pages or to exclude certain pages from the automatic exploration. *(Only links matching the list of extensions set in the advanced preference panel are explored in a Dig. Some link types are also systematically filtered out from the exploration: log-out pages, feeds which cannot be opened by the browser, etc.)*

- **Slideshow** When checked, the program will display a slideshow of the images found in the current page (or of all browsed pages, if Browse is checked).
- **Fast Scraping** In this mode exploration will be made without loading the full content of each pages, but only their textual part. This allows a much faster processing of large series of pages. In the Pro version, when Fast Scraping is checked, the Browse, Dig and Slideshow checkboxes are disabled, as well as all extractors except for Scrapers.
  The *Expert & Enterprise editions* allows you to *Fast-Scrape* and *Dig* at several levels of depth. If you check *Fast-Scrape* and *Browse*, the program will not automatically look for the next page link, as it does when Fast-Scraping is off, but it will use the scraper queue of URLs to visits that you can populate with the #addToQueue# directive.

  Note that browsing/digging and fast scraping are very different: applying a scraper by loading a page and going to the scraped view does the extraction from the source code of the loaded page, whereas using the 'Fast Scraping Mode' sends a query (an XML HTTP Request) for each URL, but doesn't really load the pages (ignoring images etc.). Most of the time, the result is the same, the Apply Scraper function being simply faster (hence the name). In some cases, however, the result can be different, or the Fast Scraping mode can even completely fail: the reason is that in the normal mode, events can happen that dynamically alter a page (mostly due to the execution of javascript scripts). These dynamic changes will not occur in the Fast Scraping mode, as scripts are not executed. This means that dynamically added information, javascript redirections, page reloads... will simply not happen in Fast Scraping Mode. If you notice this kind of behavior, the best way is to accept the slower method and browse through the URLs, doing the scraping page after page.

'Catch Export' Zone

- **Saving and Emptying the Catch before execution** Depending on the checked option(s), the content of the Catch can be emptied (and optionally saved to a backup file in the destination folder) before the execution of the macro. Note that it will not be restored after the execution, which means that you will have to load the backup file if you wish to recover the Catch content.
- **Export** When checked, the content of the Catch will be exported to a file after the execution of the macro. The destination file format can be defined in the Export popup menu. The export file will be saved in the destination file folder (or FTP server in Expert and Enterprise editions) that you have selected in the Destination Files zone or in the current default folder if no destination folder is defined.

'Destination Files' Zone

This zone allows you to set the location (on a local drive or on a **FTP server** if you have the *Expert or Enterprise edition*) and the name format for the files generated by your different extraction processes.

Important Note: If you set login credentials to upload the exported data to an FTP server with the 'ftp' protocol (Expert or Enterprise), the login and password will be sent in the clear on the network. *To avoid retyping the ftp address for each macro you create, you can set a default value in the advanced preference panel.*

- **Save extracted data to**: If you set an export destination for one or several extractors in the macro or for the Catch, the generated files will be saved to the destination folder (or FTP server on Expert & Enterprise) that you set here. For the case where a file of the same name already exists, the popup menu will allow you to set the suffix to add to the filename.
- **Split into several files**: This checkbox and menu allow you to ask OutWit Hub to close the current export file and create a new one when the chosen condition becomes true. In the current version, the only criterion available is the number of extracted rows.
- **Save downloaded files to**: If you have instructed one or several extractors to download selected files, these will be saved in the folder you set here. For the case a file of the same name already exists, the popup menu will allow you to set the suffix to add to the filename.

'Extractor Setting' Panel

This panel contains the list of all extraction processes that can be performed during the execution of your macro.

- **Extractor** column: In this column, check the extractors that you wish to activate during the execution of the macro.
- **Empty** column: When checked (like in the views bottom panels), the data resulting from the extraction of one page will be cleared each time a new page is loaded. *Note that you should not uncheck 'Empty' if you automatically move the data to the catch as the operation will be executed each time a new page is visited, with a number of rows growing exponentially.*
- **Options** column: You can set, in this column, options that are specific to each extractor. These options correspond to the ones you will find in the bottom panel of each respective view in the Hub.
- **Select If** column: Allows you to set the criterion that will trigger the actions defined in the destination column (moving the selected data to the Catch, downloading files, exporting the data to a file). Like in the views bottom panels, you can set a selection criterion on either a specific field of the extractor or on all fields.
- **Sorted by & Limit to** columns: In association with the limit to option, the sort option allows you to set the sort column (field) and direction, in order to only extract the first n rows of extracted data.
- **Destination** column: Allows you to set the destination of the extracted data and files:
  - **Catch**: When checked, moves the selected data to the Catch (which can be exported to a file at the end of the process). *Note that you should not uncheck 'Empty' if you automatically move the data to the catch as the operation will be executed each time a new page is visited, with a number of rows growing exponentially.*
  - **Download**: When checked, tries to download all files corresponding to URLs found in the selected data to the set destination folder.
  - **Export**: Exports the selected data to a file in the Extracted data destination folder (or FTP server on Expert & Enterprise) set above. You can set the export format using the Export popup menu.

IMPORTANT NOTE: *In general, you should choose* ***ONLY ONE*** *of these settings at the same time:*

- *Uncheck 'Empty' to keep the data in the view,* ***OR***
- *Check 'Catch' to move the data to the Catch (and export it at the end of the process),* ***OR***
- *Check 'Export' to export the data to a file during the process.*

The *Jobs* View *(Pro, Expert & Enterprise Editions)*

***Gives access to the Job Manager and Editor*.**

The *jobs view* displays the job manager and editor as two panels of the same window.

A Job is an automator that allows you to set the date and time at which a series of actions (**workflow**) should be performed and, optionally, the periodicity at which these actions should be repeated.

In the current version, two types of tasks can be programmed in a job: your **macros** and a list of preset **actions** for managing directly the application interface (restart, pause, change view, etc.). If you are running the *Expert & Enterprise editions*, you may also include system **commands** (Windows, Linux or OSX) in your workflow.

To create a Job, click on the 'New' button, select your macros, actions or commands in the Workflow section of the editor (adding as many lines as required), set a time and periodictity, then click on the 'Save' button.

In order for a Job action to be executed, OutWit Hub Pro must be running. If you quit the application before the time of execution, the job tasks will not be performed. However, when you start the application again, OutWit Hub Pro will check if any recurring jobs have been missed. A preference (Tools>Preferences>Advanced) allows you to decide if, in this case, you want the application to ask you whether you wish the program to execute the job(s) immediately (possibly aborting already running processes), cancel all future executions or wait for the next scheduled execution time.

Deactivated jobs will not be executed. You need to check a job's Active box in the manager, in order to program its execution. A job is always inactive if it is scheduled for a time in the past and if no periodicity is set.

In the **scheduling** part of the editor, you can either set a precise execution time or a delay to wait until the execution. You can also set a (fixed or random) periodicity and the maximum run time allowed for the job execution.

- **Periodicity** is the time elapsed between two runs (measured between execution starts), you can set it in minutes, hours or days.
- **Randomization** will add a random value (between 0 and the value you set), to the chosen value of the periodicity.
- Finally, the **maximum execution time** creates a parallel timer, which will force the job to stop after the chosen time. *(This feature was added for periodically refreshed pages which can, in some cases, prevent a job to end. It is generally not needed unless you experience a specific problem.)*

If you right-click on the name of a job in the manager (left side), the contextual menu allows you to:

- **Delete** Deletes the current job from your profile.
- **Duplicate** Duplicates the current job.
- **Export** Exports the job as an XML file or as a User Gear file (.owc) to your hard disk.
- **Properties** Displays the Properties Dialog where information about the current job can be found and edited (name, author, comments, etc.).
- **Execute** Executes the job.
- **New** Opens a new blank job.
- **Import** Loads a previously exported job.
- **Assign to Project** Adds the selected job to a new or existing project.

The *Queries* View *(Pro, Expert & Enterprise Editions)*

***Constitute and manage directories of links or query strings*.**

Allows you to constitute and manage directories of links, strings and *URL generation matrices*, in order to perform search and extraction tasks from multiple sources, in a single process.

The query directory manager is located on the left side of the window and the directory content editor on the right.

Send To Queries... In all datasheet panels, the right-click menu allows to send Link(s) or Cell(s) to any existing directory of the queries view or to create a new directory with the selection. Sending a cell or a multiple selection of cells means sending the content of the selected cell(s) in the selected column. Sending a Link or a multiple selection of Links to the queries means that the first URL (ignoring the Source URL) of every selected row will be stored in the Query String column. If no link is found in the row, no Query String will be created for this row.

You can also **populate queries using drag and drop**:
- on a directory name in the manager, the content will be added to the directory;
- in the empty part of the manager or on the 'New' button, a new directory will be created with the dropped content;
- in the directory content, the content will be inserted in the directory, at the drop point.

If you want to **type the queries manually**:
- Create a new directory by clicking on the 'New' button,
- Right-click on the right part of the screen and select 'Insert Row' to enter one URL, or 'Insert Rows' to generate a series of queries,
- You can also right-click on the right part of the screen and select 'Edit>Paste' or use the shortcut (crtl-V for Windows and Linux, cmd-V for Mac) to paste URLs that you have copied from any other application,
- Then, to edit a cell, just double-click on it.

In the *Expert & Enterprise editions*, you can **generate query directories directly from a *scraper***, using the #sendToQueries# directive.

If you right-click on the name of a query directory in the manager (left side), the contextual menu allows you to:

- **Delete** Deletes the current directory from your profile.
- **Duplicate** Duplicates the current directory.
- **Export** Exports the directory as an XML file or as a User Gear file (.owc) to your hard disk.
- **Properties** Displays the Properties Dialog where information about the current directory can be found and edited (name, author, comments, etc.).
- **New** Opens a new blank directory.
- **Import** Loads a previously exported directory.
- **Split Directory** Splits the directory content into three or more new directories. Each one will contain a maximum of 100,000 queries. (The original directory will remain unchanged.)
- **Assign to Project** Adds the selected directory to a new or existing project.

Double-clicking on the name of a query directory allows you rename it.

When you open a text file (File>Open) which only contains well formed URLs, the program will propose to send them directly to the queries view, using the name of the file for the new directory.

To use a directory of URLs, you can either select the URLs you want, right-click on one of them and choose an Auto-Explore function, or refer to it in a macro, using 'queries/YourDirectoryName' in the Start Page field of the *macro editor*.

Note that, depending on the preference setting, double-clicking on a URL may take you to the Page view and load the URL. If you do not like this behavior, you can change the preference. You can also edit the cell by right-clicking on it and choosing 'Edit Cell'.

Query Generation Patterns *(Pro, Expert & Enterprise Editions)*

OutWit's simple string sequence generation format.

This function allows you to generate a series of character strings using a simple pattern *(or generation matrix)* where variable or incremented parts are expressed between brackets.
The most frequent use of this function is the generation of URLs that will be used to execute queries from online sources (hence the name).

This feature is used in several parts of the program:

- patterns can be used instead of URLs, in the directories of the *queries* view (which, in turn, can be used in macros),
- they can be typed directly in the Start Page field of the *macro editor*,
- they can be entered and edited in the ***String Generation Panel***, using the *datasheet right-click option* 'Insert Rows...' in any datasheet of the application.

What are String Generation Patterns?

Imagine you wish to scrape a database, wikipedia pages or search engine results for the birth date of musicians (say: Wolfgang Amadeus Mozart, Franz Liszt, Johannes Brahms, Frederic Chopin).

You can, for instance, want to visit the following pages:

- http://www.myMusicDB.org/search?Wolfgang%20Amadeus%20Mozart
- http://www.myMusicDB.org/search?Franz%20Liszt
- http://www.myMusicDB.org/search?Johannes%20Brahms
- http://www.myMusicDB.org/search?Frederic%20Chopin

With OutWit, you will be able to define the queries as a single pattern string:
http://www.myMusicDB.org/search?[Wolfgang Amadeus Mozart;Franz Liszt;Johannes Brahms;Frederic Chopin]

or, for a series of pictures:
myFavoriteMovieStar[001:200].jpg,
which will generate the 200 names in the range.

NOTE: If you haven't used %20 in the pattern and if the Hub recognizes that the feature is being used to generate URLs (http/ftp/file...), space characters will be replaced by +.

Types of ranges:

Numerical Increments:

Use **[xx:yy]** to define the range of numerical indexes to increment. (If you put leading zeros, OutWit will add leading zeros in the whole series.)

Example:

string[5:11] will generate the following sequence: *string5, string6, string7, string8, string9, string10, string11*
string[095:101]will generate the following sequence: *string095, string096, string097, string098, string099, string100, string101*
string[312:309] will generate the following sequence: *string312, string311, string310, string309*

Alphabetical Increments:

Use **[aa:bb]** to define the range of alphabetical indexes to increment.

Example:
string[A:D] will generate the following sequence: *stringA, stringB, stringC, stringD*
string[auz:avb] will generate the following sequence: *stringauz, stringava, stringavb*
string[z:u] will generate *stringz, stringy, stringx, stringw, stringv, stringu*

Series of Values

Use **[one;two;three;four]** to define a series of values.

Example:

"**[jan;feb;mar;apr;may;june], [01:31]**" will generate all combinations between "*jan, 01*" and "*june, 31*" (Yes, it will create 31 days for each month.)

Step

Use **[xx:yy/5]** to define a step of 5 between the generated values.

Synchronized series and ranges

Use **[#n#a;b;c;d;e]** to mark groups of series that must be incremented together item by item, without generating all possible combinations:

Examples:

firstName=[#1#peter;mike;john;martin]&lastName=[#1#black;wilson;moore;woods] will only generate:
firstName=peter&lastName=black
firstName=mike&lastName=wilson
firstName=john&lastName=moore
firstName=martin&lastName=woods

fromYear=[#1#1960:2000/10]&toYear=[#1#1969:2009/10] will generate:
fromYear=1960&toYear=1969
fromYear=1970&toYear=1979
fromYear=1980&toYear=1989
fromYear=1990&toYear=1999
fromYear=2000&toYear=2009

If the series of a group have different sizes, the program will roll as many times as necessary through the values of the smallest to match the number of values of the longest.
[#1#Sylvie;Mary;Georgio;Gus;Rita;Paulo;Stef;Lily]-
>[#1#Blue_Room;Red_Room;Pink_Room] will generate:
Sylvie->Blue_Room
Mary->Red_Room
Georgio->Pink_Room
Gus->Blue_Room
Rita->Red_Room
Paulo->Pink_Room
Stef->Blue_Room
Lily->Red_Room

Variables:

Prompt:

Use **[#ASK#]** if you wish to be prompted for a value at the time of resolution of the pattern. (Do not use this option if the pattern is to be used in a job that you wish to execute in your absence.) Note that you can use the OutWit Query Generation format in the string you type here.

Time Variables:

Use **[#YEAR#]**, **[#MONTH#]**, **[#DAY#]**, **[#HOURS#]**, **[#MINUTES#]**, **[#SECONDS#]**, **[#MILLISECONDS#]**, **[#DATE#]**, **[#TIME#]**, **[#DATETIME#]** to insert the respective values in your string.

Example:
www.myCompany.com/monthlyReport/[#YEAR#]/[#MONTH#]/ will generate the following string: *www.myCompany.com/monthlyReport/2013/09/* (... if executed in September 2013).

Random:

Use **#RANDOM[xx:yy]#** if you wish to generate a random integer. This will only generate one integer. If you want a series of random numbers, you can use: **[#RANDOM[xx:yy]#;#RANDOM[xx:yy]#;#RANDOM[xx:yy]#;...]**. (Note that this feature only works with digits at the moment.)

Use **[#SHUFFLE#xx:yy]** or **[#SHUFFLE#a;b;c;d;e]**if you wish to shuffle the list of generated strings. This is useful, for instance if you generate URLs to explore and want to do it in a random order instead of sequentially, or if you want to shuffle the list of proxy IP addresses you entered in the proxy preference panel.

Escaping Characters:

In general the simplest way to avoid syntax errors is to filter out the characters used in the pattern syntax (mostly colon, semicolon, left and right square brackets). If you have to use one of the reserved characters in a pattern, escape it as you would for a *regular expression*. This means that if you need to use one of the following characters **; : [** or **]**, you need to type a **** before it.

POST Queries *(Pro, Expert & Enterprise Editions)*

OutWit's simple format to store and send POST queries for filling online forms.

POST queries are used in Websites to send info to the server without making it visible in the URL. It is usually sent by a form when you click on the 'Submit' button. (A form is not always a page with many textboxes and a 'Submit' button; it can be a login page, a single popup menu, or just a button in the page...).

This function allows you to include POST parameters directly in a URL to be used by OutWit, thanks to a simple format.

This feature can be used everywhere in the program:

- in the address bar,
- in the Home Page box of the General Preference Panel
- in a directory of the *queries* view (which, in turn, can be used in macros),
- in a generation pattern,
- in the Start Page field of the *macro editor*,
- in the generation pattern in the *datasheet right-click option* 'Insert Rows...'

What are POST queries?

POST is one of the request methods in the HTTP protocol used on the Web. The POST request method is used when the browser needs to send data to the server as part of the request, such as when submitting a completed form. In contrast to the GET request method where only a URL and headers are sent to the server, POST requests also include a message body. This allows for arbitrary length data of any type to be sent to the server.

OutWit's POST query format

This format is a way to simply include these POST fields (hidden or not) directly in a URL. For this, you just need to use the following syntax:

```
http://www.mySite.com/mySearchPage?#POST#field1=value1&#POST#field2=value2...
```

As you can see, the field values are simply added using the common URL parameter syntax ?...=...&...=...& etc. where all the parameters which must be passed as form fields should be prefixed with the string "#POST#". URLs like this can then be put in a directory of the queries view and used as any other in OutWit Hub.

Four other keywords can be used to set elements of your query: If you use "#HEADER#" instead of #POST the prefixed parameters will be part of the request's header, #CHARSET# will define the character encoding, #TYPE#, the contentType and #REFERER# or #REFERRER#, the request referrer.

Depending on the case, POST queries are more or less easy to use. Sometimes the form just needs values for the fields you can see. In this case using the above format is enough. But in other pages, there are hidden fields that also need to be filled. You need to know what the data should look like for your particular form (hidden fields?).

With OutWit Hub (*Expert & Enterprise editions*)
For this, the easiest solution is in OutWit Hub Expert: load the page with the form you wish to fill and before clicking on 'Submit', 'OK' or doing the action that will validate your action, choose **Intercept Next POST Query** in the Navigation menu. Then you can fill the fields you want and click on 'Submit' or other in your page. A dialog will appear with the POST query, using the POST OutWit format. You can copy it, save it in a directory of the 'queries' view, paste it in the URL bar or in the start page of a macro, etc. When the dialog is displayed, you can directly edit the query in the text box and send it. If you click on 'Cancel', the query will not be sent to the server.

Another tool of the Navigation menu is **Replace In Next POST Query**: When selected, this menu item opens a dialog which allows you to set one or several string(s) to look for in the outgoing POST

query and the corresponding replacement string(s). If you need to replace several strings, separate them using semicolons.
If you enter several lines in the second field of the dialog, the replacement will be done in as many queries as there are lines, allowing you to replace strings in a whole series of outgoing POST queries.

Note: *A query that you intercept and send to the server may not produce the same effect as the original, had it not been intercepted, especially if you have edited it. There are many possible causes for this. One of the most frequent reasons is that the site owners may have introduced an encrypted key within the POST parameters. Such keys may even be time-sensitive.*
In such cases, replacements can be more reliable as they are done on the fly, leaving the other parameters untouched. These functions will help you cover a broad scope of situations and needs but there are always cases that cannot be solved. You may want to try other available tools to snif / intercept outgoing queries, in order to build your own with the OutWit POST format.

With other tools
If you do not own the Expert (or Enterprise) edition or if the information you get with the 'Intercept' function is not enough, you can use a program like the firefox add-on called "httpfox" (there are several similar tools available). Run httpfox, click on 'Start' to tell the program to begin listening outgoing and incoming data, click on the submit button of your form and look at the POST data tab in httpfox. It will give you all the fields of the POST data. Then you can create your POST URL in OutWit.

Very Important Note: *Forms filled this way are NOT SECURE at all. In a login form, for instance, the info (including the password) being included within the URL, it will remain in your browser cache.*

The *Projects* View *(Pro, Expert & Enterprise Editions)*

***Constitute and manage directories of automators*.**

Allows you to constitute and manage directories of automators and group them into projects. A project can contain, for instance, one or several scraper(s), one or several macro(s) to apply the scrapers automatically, a directory of URLs to which the scrapers should be applied, and a job to schedule the execution.

Project are destined to help you organize your automators but are not themselves executable workflows. If you wish to execute several macros in a row, you can create a multi-step task in the jobs view.

You may group your automators any way you need, depending on your workflows.

In all automator managers, the right-click menu allows to **assign an automator to any existing project** of the project view or to create a new project with the selection. Sending an automator or a multiple selection means adding the automator to the list of automators in the project. It does not alter the automator itself. An automator can be assigned to one project, to several projects, or to none.

You can also populate projects using drag and drop:

By dropping a selection of automators...
- on a project name in the manager, the automators will be added to the project;
- in the empty part of the manager or on the 'New' button, a new project will be created with the dropped automators;
- in the project content, the automators will be inserted in the project, at the drop point.

or create projects manually:

- Create a new project by clicking on the 'New' button,
- Right-click on the right part of the screen and select Add Automator and select the automator from the proposed list.

If you right-click on the name of a project in the manager (left side), the contextual menu allows you to:

- **Delete** Deletes the current project from your profile.
- **Duplicate** Duplicates the current project.
- **Export** Exports the project and all the automators it contains as XML files or as one User Gear file (.owc) to your hard disk.
- **Properties** Displays the Properties Dialog where information about the current project can be found and edited (name, author, comments, etc.).
- **New** Opens a new blank project.
- **Import** Loads a previously exported project.

Double-clicking on the name of a project allows you rename it.

The Navigation Bar

***OutWit Hub's Navigation Toolbar*.**

The navigation bar is located at the top of the window, under the menu bar. It allows you to browse manually or automatically through URLs and remains visible and active whatever view you are in. Here are the different controls of Outwit Hub’s navigation bar.

- ***Back***: Goes back one page in the navigation history.
- ***Forward***: Goes forward one page in the navigation history.
- ***Refresh***: Reloads the current page.
- ***Home***: Loads the home page.
- ***Stop***: Aborts current processes, like the loading of a page, the dig and browse functions, the execution of a macro, etc. In many instances, the escape key has the same effect.
  Only active when Outwit is browsing, digging, loading a page, etc.
- ***Next in series***: Loads the next page in a series.
  Active when OutWit finds a navigation link to the following page (i.e. if the current page is part of a series, like a result page for a query in a search engine).
- ***Browse***: Auto-browses through the pages of a series.
  Active when OutWit finds a navigation link to a following page (i.e. if the current page is part of a series, like a result page for a query in a search engine). Escape or a second click on the Browse button will stop the auto-browse process. Right-clicking or clicking and holding down the Browse button shows a menu allowing to choose the extent of the automatic browse to perform (2,3,5,10 or all pages).
- ***Dig***: Automatically explores the links of the current page.
  Active when OutWit finds links in the current page. Right-clicking or clicking and holding down the Dig button shows a menu allowing you to constrain the exploration to or outside the current domain, and to set the depth of the dig. Depth = 0 will browse through all the links of the page, Depth = 1 will also explore the all the links of pages visited. Escape, Stop, or a second click on the Dig button will stop the dig process. (Only links matching the list of extensions set in the Automatic Exploration preference panel are explored. Some link types are systematically filtered out from the exploration: log-out pages, feeds which cannot be opened by the browser, etc.)
- ***Up to Site Home***: Goes up one level towards the home page of the current site.
  Active when the current page is not the home page of a site. Goes to the top of the current site's hierarchy.
- ***Slideshow***: Displays the images of the page as a slideshow. Shortcut: ctrl+alt+S.
  Active when OutWit finds images in the current page. The slideshow can be viewed in full screen or in the page view. If the current page is part of a series, the slideshow will go on as long as a next page is found. You can choose in the preferences, the minimal width and height of the images to display in slideshows.
- **Address bar:** Text box where you can enter a URL to load, the path to a file on your hard disk, an OutWit macro, or a query that will be forwarded to the preferred search engine.
- **About:** Displays the version number and author information about OutWit Hub and the loaded Kernel.

The *Next Page* function

Tries to determine which of the URLs found in the current page (if any) is the most probable link to the 'Next Page'.

As soon as a new page is loaded, all the links are located by the program in the source code of the page. OutWit then determines if one of them is probably the next page to visit. If none is found with enough certainty the Next and Browse buttons (single and double arrows) will remain disabled. If one is selected by OutWit as the best *next page link*, both arrows become active and respectively allow to go to the next page or to browse through the whole series of pages, as long as a *next page link* is found.

> *Note: The automatic exploration of series of pages (like the auto-browse function, accessible with the double right arrow in the Hub's toolbar), is based on the recognition by the program of navigation links in each page. A brief description of the way OutWit searches for a possible next page will help understand why the feature works rather well in general but will also explain that the program is not able to recognize all page sequences:*
>
> *The program analyzes the available html source code of the page. Next page links are looked for, using many different strategies: analyzing the URLs, the link node's text, the alternate text of image links (using vocabularies in more than 50 languages), or recognizing successions of incremented URLs, etc. The program gives a rating to each possible link found and decides of the best possible answer, if any. This analysis can fail for different reasons, like unknown (or too unspecific) vocabulary, unpopulated languages, navigation links generated by scripts or obfuscated, in dynamic ajax pages for instance. OutWit's intelligent algorithms try to discover everything they can and we will make sure they become more and more efficient, but the very nature of the problem makes it impossible to ever reach a 100% success rate.*
>
> *If your need for the scraped data is critical in your workflow and if you cannot afford failed automatic recognition, it may not be a good idea to rely on automatic features like this one, the 'Guess' view, or 'Guess Contact Info' for instance. In these cases, you should create a scraper which sets the next page link using the #nextpage# directive or manually define lists of URLs to be scraped one after the other, then execute the extraction from a macro or using the right click menu option: 'Auto-Explore' > 'Fast Scrape'. This way, if you have thoroughly tested the scraper you have designed, the process will be reliable and reproducible, at least as long as the online source is not altered and remains accessible.*

The *Dig* function

***Automaticallly explores the links of the current page 'digging' to next level pages according to the set depth*.**

Active when OutWit finds links in the current page. Right-clicking or holding down the Dig button opens a menu allowing you to limit the exploration within or outside the current domain and to set the depth of the dig. Depth = 0 will browse through all the links of the page, Depth = 1 will also explore the all the links of pages visited. Escape or a second click of the button will stop the dig. (Only links matching the list of extensions set in the Automatic Exploration preference panel are explored. Some link types are systematically filtered out from the exploration: log-out pages, feeds which cannot be opened by the browser, etc.)

The *'Dig'* and *'Browse & Dig Result Pages'* options of the Navigation>AutoExplore Pages menu

Give access to the Dig sub-menu: The Dig function is a systematic exploration of all links found in a page, in a whole site or in a series of result pages. In order to not visit hundreds of unwanted pages randomly, you can set a number of limitations. You can visit pages if they are within the same domain as the current page, outside the page domain or you can visit any link found. You can also specify the *Depth* of your exploration: Depth 0 is the list of links found in the page, depth 1 also includes all the links found in each visited page and depth 2 does the same one level below. In the *Advanced Settings* dialog, you can combine all these criteria and even set an additional filter with a string (or a regular expression) which must be present in the URL for the program to explore it. *(Only links matching the list of extensions set in the Automatic Exploration preference panel are explored in a Dig. Some link types are also systematically filtered out from the exploration: log-out pages, feeds which cannot be opened by the browser, etc.)*

> *Note: Digging through a Web site can be done in* ***Fast mode*** *to apply a scraper with the Expert & Enterprise editions: Both in a macro and using a right click on the Page in the OutWit browser, the Expert/Enterprise editions can apply scrapers to the oringinal HTML source code of entire Websites.*

OutWit Hub's Main Panels

The program interface is composed of a series of views and panels which share common features.

The main panels are the side panel which includes the list of available views, the Catch and all the other datasheets of the application, the detail panel, which displays in a more legible way the selected row of the current datasheet.

The Datasheet Panels

***In each extractor view, the datasheet is the table that contains the results of the extraction*.**

Available options vary with the view and the license of your product.

Right-Click Menu

If you right click on the selected datasheet rows, a contextual popup menu will propose a number of additional features for selection, edition, deletion, export of cells, columns, rows, etc.

Editing Cells

If the content of a cell is not a URL, double-clicking on the cell will allow in-cell editing of its content. If it is a URL, double-clicking on the cell will switch to the page view and load the URL. If you wish to edit a cell containing a URL, you need to right click (ctrl-click on Macintosh) on the cell and select 'Edit Cell'.

Sorting the data

To reorder the rows of the datasheet by a column, click on the header of the column. OutWit will determine the most pertinent way to sort depending on the type of data in the column (alphabetical, numerical, by size...). The sort order is particularly important when using the limit function of the bottom panel.

Resizing Columns

To resize a column, click and drag the line between the header of the column and the following. To resize all columns automatically, double-click on a line between two headers

The Bottom Panels

***At the bottom of each extractor view, a series of controls allow you to set various extraction options*.**

Each view features a number of options and controls depending on the nature of the data extracted. The main purpose of the bottom panels' controls is to decide which data is going to be moved to the Catch or exported.

Data Selection Filters

A series of controls are available to filter and select the extracted data.

Action Buttons

The Empty and Catch buttons have two positions *Automatic* and *On Demand*:

Empty on Demand/Auto-Empty: If Auto-Empty is selected, the content of the datasheet will be cleared each time a new page is loaded, otherwise the data will remain in the view until you click on the button to clear the view or you close the application.

Catch on Demand/Auto-Catch: If Auto-Catch is selected, the selected rows of the datasheet will be moved to the catch each time a new page is loaded, otherwise the data will only be moved to the Catch when you click on the button.

Remember that the contents of the Catch is persistent, whereas the data located in the views will be lost when you close the application (and cleared by default when a new page is loaded).

Note: Use the column picker at the top right corner to show/hide them. If you want to use a hidden column as selection criterion, you will first have to show it. When a row is moved to the Catch, the Source URL is always included in the moved data, even if the column is hidden in the view datasheet. All other hidden columns are ignored in the transfer.

Filter Controls

***The bottom panel of each extractor view contains a series of filter controls to define the data selection rules*.**

Available controls and options vary with the view and the license of your product.

Select row if

The datasheet displays all data of a given kind extracted by the view. All or some of it can be selected and moved to the catch, exported, copied, etc. using the filters that you set.

Column: The first menu (*any column* by default), allows you to set the column to which the filter will be applied. Keeping the default value will search in the whole row.
Condition: This menu defines the condition to be used to determine if a row should be selected (*contains* by default).
Text to find: In this text box, enter the pattern to search, in the form of a simple string or of a regular expression (surrounded by '/').

Limit to

When a value n, higher than zero, is set in this field, only the top n items matching the selection criterion will be selected. The selection is done in the sort order defined for the datasheet: to sort the datasheet rows, simply click of the header of the desired column.

Options

This section allows you to set the options that are specific to a given extractor view. The following controls are therefore not present in all views:

Clean Text: When checked, HTML tags or partial tags are removed from cell contents as well as leading punctuations and leading and trailing spaces. Additionally, in order to be able to copy and paste the content in a spreadsheet for instance, semicolons are replaced by commas and return characters are replaced by semicolons.
Deduplicate: When checked, duplicate rows are removed from the datasheet.
Local: When unchecked, URLs linking to pages of the same domain as the current page will be ignored.
Cache: When unchecked, URLs found to be links to a cached page will be ignored.
Scripts: When unchecked, URLs found in scripts will be ignored.
Styles: When checked, URLs will be looked for in styles and stylesheets.
Backgrounds: When unchecked, background images will be ignored.
Adjacent: When checked, the program will try to find images or documents around the found URLs (pro version).

Note: Use the column picker at the top right corner to show/hide them. If you want to use a hidden column as selection criterion, you will first have to show it. When a row is moved to the Catch, the Source URL is always included in the moved data, even if the column is hidden in the view datasheet. All other hidden columns are ignored in the transfer.

The *Selector Editor* Panel

The Selectors Panel, accessible with ctrl-P (cmd-P on Macintosh) or the Selectors Button in the bottom panel of each view, allows you to create, display and modify OutWit Selectors.

Selectors are plain text queries, allowing to make complex multiple-criteria selections in the extracted data.

The purpose of this feature is to give maximal data selection flexibility to the user without requiring a query language or complex interface full of pop-up menus and buttons. Selectors can be saved and reused in different contexts. *This feature is work in progress and both vocabulary and scope will evolve. At the time of release of version 8.0, it applies to datasheet selection throughout the application.*

The Selector Editor contains three main parts: Two lists of terms surrounding the workspace where the query is edited. To build a selector, you simply need to bring the desired terms to the center workspace either by dragging them from the sides or by double-clicking on one of them. There are several ways to create your query.

While you are building the query, it is displayed as a plain english sentence at the top of the editor.

Say, for instance, that you wish to build the following query:

"Contains an email address which ends with the string '.edu'"

- You can drag **contains** to the workspace, then drag **email** onto **contains**, then **ends with** onto **email address**, then **string** onto **which ends with**, and finally double-click on **a string** and type **.edu**.
- You can also sucessively double-click on **contains** in the left list of relations, **email** in the objects on the right, **ends with**, **string**, and finally double-click on **a string** in the workspace and type **.edu**.
- ...or any combinations of the above, dragging a term to the empty workspace or to a previously added term, or double-clicking on a term in the lists to place it over the selected term in the workspace.
- An alternative to placing **string** on top of **which ends with** is to double-click on **which ends with**. It will add an untyped term (**something**) and allow you to give a value.
- Newly added terms are generic objects (i.e. **an email address**). When you double-click on such an object, you will be prompted for a specific value (i.e. **paul@site.com**). If the value you give is empty, the term will revert to its generic meaning.

You can create selectors with criteria for several fields, using the AND and OR operators.

Field FileName contains '.txt' and field 'Freq.' contains '2':
If you want to add several conditions linked with the OR or AND operators, you can drag --or double-click on-- 'All of these' (AND) or 'Any of these' (OR) followed by a relation or an object. You can also drag a second (third...) condition to an empty part of the workspace, which will insert the operator AND.

At the bottom of the editor is a panel with a popup menu of the previously created selectors and New, Save, Save as, Cancel buttons to manage your library of selectors. Selectors can be applied manually to the datasheet or automatically at every change. In case of complex queries or if the datasheet is large, you may prefer to apply it on demand.

The Export Preview Panel

The Export Preview Panel located at the left of each view, displays the data as it will be exported in the selected format.

Select the data export format using the **Format popup menu**, located at the top left corner of the panel. Immediately on the right of this menu is a checkbox which allows you to choose between the default layout and your custom layout for the current view and the selected format. Custom layouts can be created/edited by clicking on the **Edit** button, in the top right corner.

To proceed with the export in the selected format and layout, click on the **Export** button.

Export Formats

Available export formats may vary with your version, your license level and the view you are exporting:

- **HTML**: The data is exported as HTML with the default layout or the custom layout you have defined for this view. Field names, visibility, order and styles can be customized using the Layout Editor.
- **HTML Detail**: The data is exported as HTML, one line per field, in the format **label:value**. Field names, visibility, order and styles can be customized using the Layout Editor.
- **Excel**: The data is exported as an XML Excel file. The file extension can be set in the export preferences (depending on your operating system and platform, you may need to modify it). Field names, visibility and order can be customized using the Layout Editor.
- **CSV**: The data is exported as a Comma Separated File (CSV). The record delimiter can be set in the export preferences (depending on your operating system, platform and target application, you may need to modify it). Field names, visibility and order can be customized using the Layout Editor.
- **TXT**: The data is exported as a tab separated text file. Field names, visibility and order can be customized using the Layout Editor.
- **XML**: The data is exported as a standard XML file. Field names, visibility and order can be customized using the Layout Editor.
- **JSON**: The data is exported as a standard JavaScript Object Notation file (JSON). Field names, visibility and order can be customized using the Layout Editor.
- **vCard**: The data is exported as a vCard 3.0 file (VCF). Use this format if you want to export contacts to an application accepting it. Field visibility can be set using the Layout Editor. *Note that OutWit contact extractions from diverse Web pages are rarely usable as is and often require some cleaning in a program like Excel, but, depending on the source, contacts extracted from a given corporate site page, an open directory of restaurants or any other free online contact database can be clean and immediately useable. In such cases, it can be handy to export them directly to an address book*.
- **SQLite**: *(Enterprise)* The data is exported as an SQLite database file. Field names, visibility and order can be customized using the Layout Editor.
- **SQL**: The data is exported as an SQL file composed of INSERT queries. Field names, visibility and order can be customized using the Layout Editor. The table name used in the queries will be determinedin the final export by the file name you define when saving the export file. In the preview, it is shown as #TableName#.
- **SQL UPDATE**: The data is exported as an SQL file composed of UPDATE queries. This export format should be used if you already have a table in your SQL database with previously exported data and want to update previous records. The table name used in the queries will be determined by the file name you define when saving the export file. In the preview, it is shown as #TableName#. By default, OutWit will try to use the most logical Key to do the update. If you want to make sure the export is done using the key you need, you can define and use a custom layout: as for other export layouts, field names, visibility and order can be customized using the Layout Editor. To set the Key, just move the field you want to use to the first position. (Example: you have a table of products with SKUs and Prices and you want a daily update of the price for each SKU, using a custom scraper: in the SQL Update export layout of the scraped view, you should make sure the SKU field is the first non-hidden field of the custom layout.)

The Export Layout Editor

When you click on the Edit button, a new box appears, displaying the different column headers of the datasheet panel for this view. These column headers (or field names) can be dragged left or right, reordering the columns as you wish for your custom export layout. You can also double-click on a field name to edit it, or immediately right of it, to add or edit a field separation string. Finally, if you right-click on a field name, a popup menu will appear with additional functions (Hide/Show Field, Edit Field/Separator, Append/Remove Line Break).

Under the list of field names is a toolbar allowing you (when in HTML format) to define the style (font, weight, style, color...) of the selected field. When you alter these settings, you will see the updated layout in the data preview in real time.

Toolbar buttons:

- The first three buttons switch between *paragraphs*, *list* and *table* layouts.
- The following buttons allow you to set the *font*, *color*, *size*, *weight* and *style*.
- The rightmost button displays a textbox with the *css* string describing the field style.

When you are satisfied with your custom format, click on the **OK** button at the right of the panel toolbar (with the green check). If you want to discard your changes, click on the red **Cancel** button.

The Export Preview

For performance purposes, only a few dozen rows are rendered in real time. You can scroll down to see the rest of the data.

The String Generation Panel

The String Generation Panel (Insert>Insert Rows) is a String Generation Pattern editor.

In the upper part of the panel is the ***editor*** itself: a textbox where you can paste and edit *String and Query Generation Patterns (matrices)*. If you right-click in the textbox, a pop-up menu proposes a few sample formats for you to edit and experiment with.

Under the editor, radio controls allow you to select what you wish to generate (blank cells, identical strings, automatically generated strings or the generation pattern itself).

In the lower part, the ***preview box*** shows in real-time the first generated strings.

Two input boxes allow you to set the *step* of the increment and the *maximum number* of strings to generate.

When you are satisfied with your pattern, click on the **Insert** button and the generated string will be inserted in the current datasheet.

The Catch

The Catch is your collection basket. Drag to it the data and media you wish to keep.

It is located at the bottom of the screen and can be displayed or hidden using the 'Show Catch' option of the 'View' menu.

While browsing the Web, *the Catch* is where you can store all the information you wish to collect. You can do so either automatically, using the *move to Catch* checkbox, or manually, by dragging rows of data from a view, by pressing the *return* key when you have a selection in a datasheet or by clicking on the *'Catch'* button in the bottom panel of one of the views. You can then save your *Catch* as a file or export its content to different formats.

The Catch is not a mere list of bookmarks. It is a repository for all sorts of heterogeneous data. It will adapt to the type of items you store into it and, if the incoming data includes a field which is not already in your Catch, a new column will be added to receive this field.

As the quantity of data in the Catch can be very large and its nature very diverse, it is important to be able to mark it and sort it in a variety of ways. It can of course be sorted like all other datasheets, by clicking on the headers, but it also contains a series of additional fields for tracking purposes: *Rating*, *Priority*, *Collection Time* and *Source URL*.

Rating and Priority

The *Rating* and *Priority* cursors allow you to mark selected data with two indices. ***Rating*** (between 0 and 5) can be used to rate the importance of the data row, ***Priority*** (between 0 and 10) can be used to order the data finely. In the Catch as well as in the Detail Panel, the lines will be colored according to these values. These two parameters have been added for your convenience managing the extracted data, use them as you wish.

Save Incoming Files

When the Save Incoming Files checkbox is checked, each time a new row is added to the Catch, the program will search the row for the first link to a downloadable file and it will download it if found. The files will be saved to your hard disk, in the current destination folder. You can change the destination folder using the Preferences panel.

Datasheet Functions

The Catch is a datasheet and shares all the main functions with the other datasheets of the program, including the right-click functions.

Resizing Columns

To resize a column, click and drag the line between the header of the column and the following. To resize all columns automatically, double-click on a line between two headers.

The Detail Panel

The Detail Panel displays the currently selected row in a more legible way than in the datasheet.

The content of the *Detail* panel cannot be edited, but it can be copied.

Note, however, that large texts are truncated in this panel; it is therefore preferable to copy the data directly from the datasheet.

The *Log* Panel

***Displays a log of the session events*.**

The *log panel* presents a log of the most significant events that occurred in the current session.

If the panel is not visible on the screen, it can be displayed using the *View* menu or dragging down the info bar at the top of any view. In the page view, the log panel fills the part above the info bar, in the other views, the top panel can be split between the page browser and the log panel. You can drag the splitter from the right side of the window to the left, displaying the page, the log or both. The splitter becomes dark when the mouse rools over it. When the log panel is completely collapsed on the right, you can drag it or double-click on it to bring it to the left.

The panel contains the following information (the number of displayed columns depends on the panel width):

- The date when the event was recorded
- The time when the event was recorded
- The description of the event

Only a limited number of events are logged in the current version. We are progressively adding more, and our objective is to give you the best visibility possible on the program's exploration and extraction processes, with a complete tracking of the tasks being performed.

Panel Options

A *right click* on the panel allows you to perform almost the same actions as in the datasheets, in particular to clear the log contents, to save it in a file, etc.

The Preference Panels

The Preference Panels (Tools>Preferences) display your current preference settings.

The Preference Dialog is divided into five panels: General, Exploration, Time Settings, Export, Advanced.

The option labels should be self-explanatory. They vary more often than the rest of the user interface, which makes it difficult to discuss them here. We have, however, documented the functions within the interface itself with tooltips: if you place your cursor over the label of a control and wait for two or three seconds, you will see a yellow tooltip explaining the expected content for the control.

Export

Renaming Files

A preference allows you to rename downloaded files. Another one is for the data export files. The pattern entered in the corrsponding text box allows you to set the format of the new names. The currently allowed syntax for this pattern includes the following:

- **###** a series of pound signs will be replaced in the final file name by a numerical index to be incremented if several of the downloaded files have the same original name (leading zeros will be added to match the number of pound signs.
- type **[url]** to include the url of the page from which the file was downloaded: www.example.com › downloads › documents › help.pdf
- type **[domain]** to include the domain of the page from which the file was downloaded: example.com
- type **[domainName]** to include the domain name of the page from which the file was downloaded: example
- type **[params]** to include all the parameters from the url of the page from which the file was downloaded
- type **[paramN]** to include the Nth parameter of the url from which the file was downloaded: *[param1]* gives *q=which* for the url www.mySite.com/tester/test?q=which&page=2
- type **[param-N]** to include the Nth parameter starting from the end, of the url from which the file was downloaded
- type **[path]** to include the path from which the file was downloaded
- type **[pathN]** to include the Nth element of the path from which the file was downloaded: *[path1]* gives *tester* for the url www.mySite.com/tester/test?q=which&page=2
- type **[path-N]** to include the Nth element starting from the end, in the path from which the file was downloaded
- type **[ordinal]** to include the ordinal number of the page in the series of pages that are being visited in an automatic exploration
- type **[originalName]** or **[originalRoot]** to include the root (file name without the extension) of the file being downloaded. *(Note that the extension is handled by the download manager and will be chosen by the program to match the type of the file.)*
- type **[originalExtension]** to include the extension of the file being downloaded.
- type **[date],[time],[datetime],[hours],[minutes],[seconds],[miliseconds]**... to include date or time elements.

When renaming downloaded files, the program will alter certain characters if they are not valid in filenames for the operating system you are using. Forward slashes, in particular, will be replaced by a right-pointing Double Angle Quotation Mark (») or any other character set in the Path Separator preference.

Time Settings

The time settings allow you to set the timeout values for the HTTP and XHR queries sent by the program during automated processes. The first three sliders allow you to set respectively:

- the maximum time the program can wait between the moments when the query is sent to a server and the first byte of data is received from the server
- the maximum time allowed between the first data received from the server and the moment when the page is fully loaded

- the maximum time allowed after the page is fully loaded, for OutWit extraction processes on this page

The following two couples of sliders set the temporization (pace) of the automatic exploration/fast scraping processes. If minimum and maximum values are different, the program will wait for a random amount of time between these limits.

The last sliders allow you to pause the automatic application of scrapers to a series of URLs (fast Scraping) for a given amount of time after a given number of queries.