

GSA Search Engine Ranker

Script Manual

© GSA 2013 - www.gsa-online.de

The program is all based on a simple script engine that we developed on our own which could theoretically be used to submit anything to any platform you can think of. Though it might be a bit hard to figure out the meaning behind the values. This document tries to explain everything in an hopefully easy way.

The Structure.....	2
The SETUP Section.....	3
The STEP Section.....	7
Variables usable in SETUP and STEP Section.....	12
How form fields are filled.....	15
Data Extraction.....	21
A small Example.....	22

The Structure

Each engine is defined by a simple ini file that has to be placed in the "Engines" folder located in the programs installation folder.

To those of you who do not know what a ini file is let me explain it shortly. A ini file is a simple text file that you can open in a text editor like Notepad. It has a section, a variable and a value.

```
[SECTION]
variable=content
```

Everything is not case sensitive so you don't have to care if you write "[SECTION]" or "[Section]". There are basically two types of engines.

a) The ones that require an account and login.

You will have to define at least the following sections:

```
[SETUP], [REGISTER_STEP*], [LOGIN_STEP*], [STEP*]
```

b) Those who need no account and no login.

You just need the following sections:

```
[SETUP], [STEP*]
```

The SETUP Section

This section defines global settings used to identify a website or set a description. The possible values are described below:

Variable	Allowed values / Description
enabled	1 = Engine is usable from GUI (default) 0 = Engine is not usable from GUI
default checked	1 = When a new project is created, this engine will be enabled (default) 0 = This engine is not checked on a new project
engine type	Name the type of the engine e.g. Directory, Forum, Blog Comment, ... This will be used to sort it into the tree view when you edit or create a project. <i>Example: engine type=My Engines</i>
description	Just write some notes about this engine that get shown when the user moves the mouse over the engine name in the tree view. You can use <code>\n</code> to generate a line feed. <i>Example: description=My little Engine\n\n(c) GSA Supported\n\nHave fun</i>
dofollow	1 = This engine creates a do follow link 0 = This engine creates a no follow link 2 = This engines produces both, no follow and do follow links.
anchor text	1 = This engine creates links with your anchor text 0 = This engine creates links with there own anchor texts like "Homepage" or "Visit XYZ's Site". 2 = This engine might create both, anchor text links and links with there own anchor text.
uses pages	1 = This engine uses pages 0 = This engine uses no pages 2 = This engine uses might use pages or not. For some engines the link will move to page 2 or beyond and might not be visible if you visit that page later. The link is still there but on a different page. This is just an informational data and not used in any way right now.
page must have	This parameter is used to check whenever the webpage is usable for this engine or not. The content of this variable has to be present in the webpage (either pure text or html source). The variable can have multiple values separated by a

	<p> where just one has to match.</p> <p><i>Example:</i> page must have1=Powered by XYZ XYZ Powered page must have2=!not allowed to access this page page must have3=Webpage Homepage</p> <p>In the above example you see 3 variables and each of them has to match before a engine is used for that website. The ! parameter in "page must have2" means that the following should not appear on the page.</p>
url must have	<p>This parameter is used the same way as "page must have" but for the URL string itself and not for the website content.</p> <p><i>Example: url must have1=/wiki/ url must have2=!wikipedia.com</i></p> <p>In the above example we will not use this engine for URLs that contain wikipedia.com or do not have "/wiki/" in there URL.</p>
fixed url	<p>If no "search term" is used, you have to use this tag to indicate that the submission is just happening on one fixed site (usually some kind of web 2.0 site with blog creation).</p> <p><i>Example: fixed url=https://my.opera.com</i></p> <p>This is than adding the URL to the target URLs of the project and starts a submission to it.</p>
search term	<p>This is used to search for new targets on the internet with the help of search engines like google.</p> <p><i>Example:</i> search term="Powered by XYZ" "Powered by XYZ"</p> <p>The example above uses two possible search queries. You can use also encoded the search term already if you want with "%90%AF%24". This is useful for queries in some strange language. The program will pick one of the queries randomly so the order is not important.</p>
add keyword to search	<p>1 = Add a keyword from the project to the search query 0 = Never add a keyword to the search query 2 = Add just sometimes a keyword to it if it seems to be useful (default)</p>
use blog search	<p>0 = Never use blog search engines 1 = Use blog search engines (default)</p>

	2 = Use only blog search engines
extract keywords	1 = Extract keywords from the sites meta tags. These can be used later on 0 = Don't extract keywords (default)
extract keywords ignore	Ignores the keywords and will not add them. <i>Example: extract keywords ignore=blog,wordpress</i> If the meta keywords for that site are "SEO, blog, wordpress" then the program will just extract SEO from it.
posted domain check	Overwrites project settings: Avoid posting URL on same domain twice 0 = do not post any link if anything has been posted before 1 = allow to post a link again on the same domain (but only if the URL is different) 2 = special setting for tier projects that would than allow to post several URLs on the same site.
skip ext links on	Overwrites project settings: Skip sites with more than 1 = setting from project is applied 0 = setting from project is ignored
skip content on	Overwrites project settings: Skip sites where the following words appear 1 = setting from project is applied 0 = setting from project is ignored
skip url content on	Overwrites project settings: Skip sites with the following words in URL/Domain 1 = setting from project is applied 0 = setting from project is ignored
skip nofollow links on	Overwrites project settings: Try to skip creating Nofollow links 1 = setting from project is applied 0 = setting from project is ignored
skip pr on	Overwrites project settings: Skip sites with a PR below 1 = setting from project is applied 0 = setting from project is ignored
referrer	Sets a fixed referrer that is used when downloading content or submitting something to this site. <i>Example: referrer=%url%</i> This will always use your URL from the project as a referrer.
user agent	Instead of using a user agent that the program

	<p>chooses for you, you can define it here.</p> <p><i>Example: user agent=Some Legit Browser v1.01</i></p> <p>This might be useful for exploits where you e.g. can send a html link in the "user agent" that is then turned to a real link on a page.</p>
extract from email	<p>If you use email verification in the engine you might need to extract not only verification URLs, but also variables like a login or password from the emails. This can be done here.</p> <p><i>Example:</i> <i>extract from email=key,forum_password=email_pw</i></p> <p>This example will try to extract the data for key (section key) and forum_password (defined in email_pw section). Please read the chapter Data Extraction for details.</p>
whois only	<p>Very special variable that is only used for engines that submit your URL to sites that carry your domain in there URL.</p> <p>0 = Not a whois only engine (default) 1 = Only whois submission and if not detected as a whois one, skip submission</p>

The STEP Section

There are 3 types that can be used:

- a) [REGISTER_STEP*] used to create an account
- b) [LOGIN_STEP*] used to log into the site with the created account
- c) [STEP*] the actual submission process

You can define as many steps as you want but they have to be in sequence like [STEP1] [STEP2] and so on.

The following fixed variables can be used:

Variable	Allowed values / Description
find link	<p>Searches the site for a anchor text and downloads that URL to work on with. You can use more than one link text when separated with .</p> <p><i>Example:</i> <i>find link=Register Registrieren Sign up Signup</i></p> <p>The example will search for a link that has at least one of the above anchor texts.</p>
find url	<p>Searches the site for a URL that is matching one of the masks defined in the variable.</p> <p><i>Example:</i> <i>find url=*/register */register.php */signup/*</i></p> <p>You should always use <i>find link</i> and <i>find url</i> in combination so that if one fails the other is used.</p>
optional find url link	<p>0 = A link has to be found else we abort the submission (default) 1 = If no link is found we continue to find a form or whatever should be done in the current submission step.</p> <p><i>Example: optional find url link=0</i></p>
alternative url	<p>If <i>find url</i> and <i>find link</i> are not finding any URL that can be used, we will generate an alternative URL.</p>

	<p><i>Example: alternative url=./register.php</i></p>
browse links	<p>This will take the variable content and browse to the site. It is just useful for directories maybe where you first have to browse to a certain category till you are allowed to start a submission.</p> <p><i>Example: browse links=%category%</i></p> <p>If your project category is defined like "Computer :: Software :: Shareware", than the program tries to surf to <i>Computers</i> followed by finding a link names <i>Software</i> and than <i>Shareware</i>.</p>
browse links from root	<p>1 = The root URL is used for "browse links" in case the start URL is a sublink. 0 = Start the "browse links" from where we are.</p>
browse links url mask	<p>Tells the program to just search for URLs matching that mask.</p>
just download	<p>1 = Download the new URL and continue with STEP2 or finish the submission if nothing else should be done. 0 = a form has to be found to start the submission (default)</p>
form id	<p>Try to find a form on the current webpage that has a ID as in the variable content. Again you can use to have multiple variations. Not many sites use a ID in the <form> tag but some do.</p> <p><i>Example: form id=signupform signup</i></p> <p>This tries to find a from with the ID called signupform or singup.</p>
form name	<p>Try to find a form on the current webpage that has a name like the variable content. If no name is used in the <form> tag, the caption of the submit button is used.</p> <p><i>Example: form name=*Sign Up *Register</i></p> <p>This tries to find a form there the name or the submit button's name is either matching "*Sign Up" or "*Register".</p>
form url	<p>Try to find a form where the submission URL would match the variable content.</p> <p><i>Example: form url=*/signup.php */signup/</i></p> <p>This tries to find a form where the resulting</p>

	<p>submission URL after pressing the submit button would match <code>"/signup.php"</code> or <code>"/signup/"</code>.</p>
form request with	<p>Some Forms get sent by ajax or some other javascript framework and would than need to send a <code>"X-Requested-With"</code> in the html header. By default nothing is sent by in the HTML header.</p> <p><i>Example: form request with=XMLHttpRequest</i></p>
optional form	<p>0 = If no form is found we will stop the submission with a failer (default) 1 = the submission process is going on (STEP2 Section e.g.) even if no form has been found.</p>
seconds to wait before submission	<p>Will wait a given number of seconds before the form is actually submitted. This is useful to not trigger some robot detection system for some engines.</p> <p><i>Example: seconds to wait before submission=12</i></p> <p>Wait 12 seconds before the form is submitted.</p>
seconds to wait before submission condition	<p>This will only delay the submission if something in the variable content is found on the webpage.</p> <p><i>Example: seconds to wait before submission condition=stop_spam_time</i></p> <p>If the word <code>"stop_spam_time"</code> is either present in the html source or the visible text we wait with the submission a given time, else we skip this.</p>
post data	<p>This is hardly used but will create a custom data that is used to submit to websites instead of using the data from <code><form></code>.</p> <p><i>Example: post data=this=that&something=else</i></p> <p>Have a look in the Pingback or Trackback engine where this is used.</p>
encode post data	<p>0 = encode the data in a proper way as used in POST protocol (default) 1 = take the data as it is without encoding anything</p>
variable must be used	<p>A form is only submitted if certain variables have been used in that form.</p> <p><i>Example: variable must be used=url,login email</i></p> <p>The above example requires a form to use the variable <code>"url"</code> or <code>"login"</code> and <code>"email"</code>.</p>

<p>add fixed data</p> <p>add fixed data condition</p> <p><i>remove fixed data</i></p> <p><i>remove fixed data condition</i></p>	<p>This will add a form field if the thing in "add fixed data condition" is found on the webpage. This works only if both variables are defined. Don't worry about this too much as you will probably never use this. It is only required if some sites add data to forms by javascript to prevent automated submissions. You can use "remove fixed data" to just remove that from the posting.</p> <p><i>Example:</i> <i>add fixed data=code=%captcha%;hidden=0</i> <i>add fixed data condition=code_sign;hidden_data</i></p> <p>If the webpage has anywhere the word "code_sign" in it we add a new form field called code that will be filled with a captcha. And if the word "hidden_data" is found we also add a new form field called "hidden" with the content "0".</p>
<p>set unknown variable</p> <p>set unknown variable condition</p>	<p>If a form field is unknown as we didn't define how to fill it in our engine, we could still fill it by something you define here. The submission aborts if this is not defined and something is unable to get filled. The "set unknown variable condition" doesn't have to be used but if it is, the string on that variable must be present to use this.</p> <p><i>Example: set unknown variable=%question%</i></p> <p>If a field is found called "blahantispam193802378" we can still define how to fill this and it gets filled by the value %question% which is handled by a popup asking you how to fill it. Also often used it "set unknown variable=%leave%" which will simply leave the value as it is.</p>
<p>match by option label</p>	<p>1 = A form with a select or radio field is filled by checking the variable content against the option labels (the one you see on the browser).</p> <p>0 = We will not check for a matching label</p> <p><i>Example: match by option label=1</i></p>
<p>match by option value</p>	<p>1 = A form with a select or radio field is filled by checking the variable content against the option value (hidden from the users eye in browser)</p> <p>0 = We will not check for a matching value</p>

<p>match by prefill</p>	<p>1 = If the user has filled this field before in the past we use that content to fill it (default) 0 = we do not check past submissions on what has been used here.</p>
<p>modify submit url modify submit url condition</p>	<p>Sometimes a javascript is modifying the actual submission URL of a form which is not recognised by default. In this case you can do this with the variables. "modify submit url condition" doesn't have to be present.</p> <p><i>Example:</i> <code>modify submit url=./submit.php?changed=1</code> <code>modify submit url condition=javascript:modify_url</code></p> <p>The form URL is changed to ". /submit.php?changed=1" if the string "javascript:modify_url" is found on the webpage.</p>
<p>remove cookie remove cookie condition</p>	<p>Removes a cookie manually for the given URL/domain</p> <p><i>Example: remove cookies=%targeturl%</i></p> <p>This removes any cookie for this domain</p>
<p>set cookie set cookie condition</p>	<p>Sometimes cookies are required to submit a form and they are set by javascript so that the program can not detect them. You can do that manually with these variables. "set cookie condition" doesn't have to be present.</p> <p><i>Example: set cookie=passed_check=1</i></p> <p>This sets a cookie named "passed_check" with the value "1".</p>

Variables usable in SETUP and STEP Section

Some variables can be used globally (SETUP) or in the STEP (some only on STEP1) section. not

<p>submit success submit failed captcha failed</p>	<p>At the end of each STEP* the resulting webpage content is checked first against "submit success" and than against "submit failed". And if one is matching the submission is either set as successful or failed. The text is not case sensitive and multiple items can be added with .</p> <p>Example: submit failed=<div id='errorbox'> submit success=submitted successfully captcha failed=wrong code letters wrong</p> <p>If the text "submitted successfully" appears on the webpage (source or visible text) then the submission is taken as successful. If the text "<div id='errorbox'>" is found then the submission is aborted.</p>
<p>submit success skip verify</p>	<p>If a submission is detected as successful and it is e.g. a REGISTER_STEP* then we can skip the verification process if certain stings appear on the page.</p> <p>Example: submit success skip verify=>Logout< >Log Out<</p> <p>After a successful registration we would normally wait till an verification email arrives to continue with login and submission. However some platforms might allow you to login without this or log you in already. In that case it would be a waste of time to wait for a email from them so we continue to login and submit the content.</p>
<p>verify on unknown status</p>	<p>1 = if a submission is not detected as successful or failed it will still be taken as successful (appearing in log with "unknown submission status") 0 = we assume the submission failed (default)</p>
<p>verify submission</p>	<p>1 = verify the submission 0 = do not verify the submission but assume that the link is submitted and will be visible there or is already (default) Even though this is the default behaviour, you should set it to "1" as used in most engines. 2 = this submission step is not creating a link but is used for something else. Example: verify submission=1</p>

	<p>After a successful submission the URL is put to the verification list and checked in intervals if the link appears there.</p>
<p>verify by</p> <p>verify search for</p>	<p>Defines how to verify a submission. Possible value for "verify by":</p> <p>search = try to locate the search form on the page and submit a search using "verify search for"</p> <p>url = downloads the URL in "verify url" and checks for the present of "verify search for" which is by default your URL</p> <p>txturl = same as "url" but checks also if the link is appearing just as a text (no html link code).</p> <p>email = logs into your email account and tries to find a link that is coming from the same webpage domain.</p> <p>extractemail = just parses for data in emails without downloading any URL. This might be important for verification URLs that are used later on in scripts.</p> <p><i>Example:</i> <code>verify by=email+search,url</code> <code>verify search for=%website_title%,%url%</code></p> <p>This will do actually two verifications (separated by ,). First it tries to log into your email account and checks for links and also checks on the webpage itself using the search form and searching for "Website title". Once found by one of the methods it will use that resulting URL to check for your URL.</p>
<p>verify url</p> <p>verify url remove</p> <p>verify url replace</p>	<p>If you use "search", "url" or "txturl" in the variable "verify by" and you don't want to check on the last used URL then you can define a different URL here.</p> <p><i>Example:</i> <code>verify url=./index.php</code> <code>verify url replace=/edit/;/show/</code> <code>verify url remove=sub/</code></p> <p>Imagine the last URL was: "http://www.something.com/sub/edit/?a=2" It would now be "http://www.something.com/show/index.php"</p>

use original url to verify	<p>1 = this will not use the last URL but the URL we started the whole engine with. 0 = use the last URL at the end of the submission (default)</p> <p><i>Example: use original url to verify=1</i></p>
verify interval	<p>Defines in what interval in minutes this verification should take place (default 180).</p> <p><i>Example: verify interval=60</i></p> <p>This checks each hour if the submission is verified.</p>
verify timeout	<p>Defines how long we wait in minutes till we assume that a submission will never be successful and skip the whole site (default 7200).</p> <p><i>Example: verify timeout=180</i></p> <p>Waits 3 hours for a successful verification before giving up.</p>
first verify	<p>The first verification would start 10 minutes after the submission unless you define it differently.</p> <p><i>Example: first verify=60</i></p> <p>This means to check one hour after the submission.</p>
verify search detail url	<p>1 = If a URL has been located with your wanted data (URL) on it then the verification would normally be finished. Though sometimes you end up with a link not as good as it can be. This is especially the case if you use "search" in "verify by". In this case a better URL is located by seeing if a DETAIL or MORE or alike link is showing your URL as well. This is by default enabled. Set it to 0 if you think the resulting URL is the best it can find.</p> <p><i>Example: verify search detail url=1</i></p>
try to continue without verification	<p>0 = follow exact verification steps (default) 1 = try to skip verification and continue</p> <p><i>Examble: try to continue without verification=1</i></p> <p>When used in a REGISTER_STEP1 section, this would</p>

	skip e.g. the email verification and try to login and submit your link. In case this goes wrong there is still the normal way to go on (wait verification + login + submit).
modify url	This is used to change a found URL to something else. Example: modify url=%targethost%%targetpath% If the program starts the submission with a URL like " http://www.something.com/guestbook.php?page=23 " you will most likely not find a form to submit your link as this is only on page 1. So in this case the "modify url" removes all parameters from the URL and you end in " http://www.something.com/guestbook.php ".
modify url remove	The same as "verify url remove" but used to modify a URL for the submission.
modify url replace	The same as "verify url replace" but used to modify a URL for the submission.
Download retries	Number of tries to submit or download something (default is 1).
Link type	Defines the type of backlink created. Can be anything you want but you might want to use the types already used in other scripts.

How form fields are filled

First you would have to get all form field names from a form you want to submit to when looking in the html source like...

```
<form action="submit.php">
  <input name="homepage" id="hp" type="text">
  <textarea name="comment" id="cmd"></textarea>
  <input name="a1" type="hidden" value="0">
  <input name="button" value="Submit" type="submit">
</form>
```

On the example above you would take just "homepage" and "comment" as the rest is either hidden or a button that is not possible to

fill out.

```
homepage=%url%
comment=%blog_comment%
```

As you can see we fill the form field "homepage" with the content of the variable "url" and the field "comment" with the variable "blog_comment". Of course the program still needs to know what URL or Blog_Comment is so we have to make that visible for the customer to fill it out when editing a project. And this would look like the following:

```
[Blog_Comment]
type=memo
allow html=You can use these tags:|<a href="" title="">
allow return=1
must be filled=1
hint=Use some comments here that will be posted to the blogs.
default=%file-comments.dat%
html to bbs=BBCode format allowed
custom mode=1
auto modify=1
auto add anchor url=1
html to markdown=Markdown</option>
```

The list below explains all possible variables you can use here.

Variable	Allowed values / Description
type	Can be one of the following memo multiple line text simple edit box (default) email simple edit box but should have a valid email syntax

	<p>url simple edit box but should have a valid url syntax</p> <p>category a category to fill out</p> <p>file the customer has to choose a file.</p> <p>login a login field</p> <p>password a password field</p> <p>extract a special field not visible when editing but used to extract data from emails or webpages.</p>
allow html	<p>0 = No html allowed</p> <p>1 = html is allowed</p> <p><text> = If the text appears on the webpage we will allow html.</p> <p>HTML will be removed from your defined content if not supported by the engine (transformed into text only).</p>
allow return	<p>0 = no return/line feed allowed</p> <p>1 = return/line feed is allowed</p> <p><text> = If the text appears on the webpage we will allow a return/line feed.</p>
must be filled	<p>1 = User must fill this out in project data</p> <p>0 = User can leave this empty (default)</p>
hint	A text that is displayed when the user moves his mouse over the field.
default	The default value that is used when creating a new project. You can also use "%file-FILENAME%" here in case you want to load the default content from a file.
html to bbs bbs supports	<p>0 = No conversion (default)</p> <p>1 = Convert html code to BB code.</p> <p><text> = Convert to BB Code only if the text is appearing on the webpage.</p> <p><i>Example: html to bbs=1</i> <i> bbs supports=url,b,i,u</i></p> <p>The variable "bbs supports" defines what this engine can convert. Usually "url" (default).</p> <p>Please note the spelling as it is "html to bbs" and not "html to bb" as it should have been called.</p>
custom mode	<p>1 = User is not asked to manually customize this data before submission (when custom mode is on)</p> <p>0 = No customization (default)</p>
auto modify	1 = Modify the content a bit with random line feeds and other things to make it a bit more

	<p>unique (not duplicate content).</p> <p>0 = Do not modify this at all</p> <p>2 = Modify only line breaks after full sentences</p>
<p>auto add anchor url</p> <p>auto add anchor url content</p>	<p>1 = Add a link to the end of the post in case the site supports html and no URL could get posted anywhere else.</p> <p>0 = Do not add a url at all.</p> <p><i>Example:</i></p> <pre>auto add anchor url=1 auto add anchor url content=Click %anchor%</pre> <p>The program is trying to locate the anchor in the text if there is no URL used in the form. If found it will place a link there, else it will add the line from the "...content" variable and put your link there at the end. The program will use it's internal variations if you leave the "...content" variable empty.</p>
html to markdown	<p>1 = Convert html code to markdown code</p> <p>0 = Do not convert it</p> <p><text> = Convert it if the text is appearing on the website.</p>
<p>html to custom link format</p> <p>custom link format</p>	<p>1 = convert html code to a custom format</p> <p>0 = Do not convert it.</p> <p><text> = Convert it if the text is appearing on the website.</p> <p><i>Example:</i></p> <pre>html to custom link format=1 custom link format=[%url% %anchor_text%]</pre> <p>This would be used for Wiki engines where a link is encoded in a special way.</p>
<p>html line break</p> <p>html line break format</p>	<p>Converts a normal line break to some html line break (default
). You can specify the line break format in "html line break format" if it is some special line break as seen in some wiki engines.</p> <p><i>Example:</i></p> <pre>html line break=1 html line break format=<
></pre>
alternate data	<p>Right now only used for anchor_text which will use something else than the entered data according to its value</p> <p><i>Example:</i></p> <pre>alternate data={click here my page webpage}</pre>

	This will not use the anchor text that the user entered but something else randomly.
static	<p>1 = marks this as an important variable that should be saved as it might have to be used later on.</p> <p>0 = do not save the content.</p> <p>2 = saves it till next submission process is successful (e.g. for first time verifications it has to be removed later to allow further submissions)</p> <p>This is important for logins and passwords as the content could be random but we still need to know the login data for later login after a verified registration.</p>
max length	The maximum length of the resulting data.
min length	The minimum length of the resulting data.
alpha	<p>Used when the type is a login.</p> <p>0 = no alpha (A-Z) chars allowed</p> <p>1 = alpha chars allowed (default)</p>
numeric	<p>Used when the type is a login.</p> <p>0 = no numbers allowed</p> <p>1 = numbers allowed (default)</p>
upcase	<p>Used when the type is a login.</p> <p>0 = capital chars are not allowed</p> <p>1 = capital chars are allowed (default)</p>
lowcase	<p>Used when the type is a login.</p> <p>0 = lowercase chars are not allowed</p> <p>1 = lowercase chars are allowed (default)</p>
capitalize words	<p>0 = no modification</p> <p>1 = Will Capitalize Each Word</p>
remove	Remove certain content before we apply it to the form field. See "verify url remove" for syntax.
replace	Replace certain content before we apply it to the form field. See "verify url replace" for syntax.
allow spin	<p>1 = User can use spin syntax in project editor (default)</p> <p>0 = User is not allowed to use spin syntax</p>
tier data	<p>Sets the data that will overwrite the input data for tier projects. Just if the "tier data" is empty it will use the one that the user set.</p> <p><i>Example: tier data=%tier_title%</i></p> <p>If the project is a tier project, it will not use the input data, but instead fill it from the</p>

	variable %tier_data% which can e.g. be a extract-variable that downloads the URL and extracts the <title></title> value.
--	--

Of course some form fields doesn't need to get filled by a previously defined content from the user in project options. These are e.g. captcha forms or check boxes with someone has to agree to there terms. Just add the content without using variable content like this:

agree_tos=1

1 = check the box, 0 = don't check

captcha_code=**%captcha%**

%captcha% indicates that this is a field with a captcha image to fill out

text_question=**%question%**

%question% indicates that this is a random question that the user has to answer.

cap_text=**%capquest%**

%capquest% can be both (captcha or question)

country_box=**%random_option%**

Choose a random item out of a selection box.

signature={Some|A} {spin|spun} text

Just fill that field with the text after spinning it.

letters=**%textcode%** <front>;<back>

tries to fill the field by locating the data between <front> and <back>

something_else=**%leave%**

Leaves this field to it's default content/check state.

If a form field variable is not found in the current section (e.g. [STEP2]) then it is checked from [STEP1] section.

If the name="xyz" is not matching, we try to match by id="xyz". In the above example you could also define:

```
[STEP1]
hp=%url%
```

Data Extraction

Sometimes you need to extract certain content from the website or the email like a login/password. This can be done by defining a new variable type as the following

```
[email_pw]
type=extract
front1=Your password is:|Ihr passwort lautet:
back=\n
```

The "type=extract" tells the program to extract data from the email or webpage and not ask the user to fill it in project data when editing a project.

The variable "front1" shows what should be before the things we want to extract and "back" (yes you guessed it) the thing it should cut off the extracted data. You can again add multiple items with | and if you have more than one "front1" you can also add "front2=", "fornt3=" and so on. Same for "back".

Variable	Allowed values / Description
front or front1, front2...	Defines the content to cut in front of our to extract data. Separate multiple variations by .
back or back1, back2,..	Defines the content to cut in the back of our to extract data. Separate multiple variations by .
remove html	0 = Do not modify it 1 = remove html code

must have	If the extracted data has not the something appearing in "must have" we set it blank again.
increase	If the extracted data is a integer we can increase or decrease (negative number) the data by the given value.
default	If we have been unable to extract something, we can set a default value here.
url	Instead of using the current URL for extracting something, we extract the content from the given URL.
search url	Instead of using the current URL for extracting something, we search for an URL by the given mask and if found use that URL's content to extract data from.
find link find url	Instead of using front/back we extract URLs by the given parameters.
name_front / name_back value_front / value_back	Instead of using front/back we extract actually two values here and later join them with "name=value". This is useful for cookie extraction.

A small Example

In that example the search engine would try to find a new target with the search "Powered by XYZ" or "XYZ Powered". If the found site also has one of the strings on its page we use this engine and start to submit the data right away as no "REGISTER_STEP1" or "LOGIN_STEP1" is defined.

We first start to locate the correct sublink by searching for a link named "Submit your Site" and/or the URL containing "/submit.php" in it.

On that URL we search for a form that has an ID named "Submitform", has a name "*Submit" and/or has a URL which matches a mask "*/submit.php?step=2". Once found we start to extract all form fields and start to fill them.

We find "name" and fill it with a random name from the file "names.dat".

We find a form field called "captcha_image" and ask the user to fill that out.

We find the field called "url" and fill it with our URL defined in the project settings.

Now we submit the form and check if the submission is either successful (submit success) or not (submit failed).

If it was successful we will add the URL (modified to /latest.php) to our sites for later verification. There we will always download the website and check if our URL is appearing on it. If so we have a verified submission.

[SETUP]

engine type=TEST

description=Just a sample

page must have=Powered by XYZ|XYZ Powered

search term="Powered by XYZ"|"XYZ Powered"

add keyword to search=0

extract keywords=0

[URL]

type=url

hint=Enter your URL here please

must be filled=1

[STEP1]

find link=Submit your Site

find url=*/submit.php

form name=*Submit

form id=submitform

form url=*/submit.php?step=2

submit success=Your submission was successful

submit failed=Your submission failed

verify submission=1

verify by=url

verify url=./latest.php

verify search for=%url%

verify on unknown status=1

name=%spinfile-names.dat%

url=%url%

captcha_image=%captcha%

© GSA 2013 <http://www.gsa-online.de/>