

Start Here

What Can I Do?

Velocityscape Web Scraper Plus+ is a web crawler, a form automation engine, a file downloader, template extraction engine, and most importantly, it is an intuitive fusion of all these. With Web Scraper Plus+, you can login to a secure website → submit a search form → crawl the results → and scrape sections and fields of a resulting html pages to rows and columns in your favorite spreadsheet or database.

Imagine if instead of having to manually cut and paste contact or pricing information from a website, you could create an extraction template to pull them all into Microsoft Excel all at once. With Web Scraper Plus+ you can.

What if you could automate any form on the web and feed data to it directly from Excel, Access, or any number of commercial databases. Perhaps you would like to submit several thousand Google Adwords at once. With Web Scraper Plus+ you can.

Maybe you would like to create an email list from a secure website that you are a member of. The Web Scraper Plus+ crawler engine automatically saves all email addresses that it runs across, so you can just login, crawl, and go.

With Web Scraper Plus+ your imagination is limit. If you can browse it, click it, cut it, or paste it, you can do it with Web Scraper Plus+. But with Web Scraper Plus+ you can do it thousands of times faster.

System Requirements

Software Requirements

MS Windows 2000, XP, or 2003 w/ Internet Explorer 6.0 or greater. To get the full benefits of the integration with Microsoft Excel, MS Excel 2000 or later is required.

Hardware Requirements

Web Scraper Plus+ is a processor, disk, memory, and network intensive application. Fortunately, it is designed to take advantage of the latest advances in enterprise server technologies. The following specifications reflect the minimum supported configuration, but you will realize significant performance benefits by increasing the capacity of your processor, disk, and memory subsystems. The processor is the first and most important bottleneck

Minimum Hardware Requirements:

Pentium II 350
256 MB RAM
1 GB Free Disk Space
128k Network Connection

Recommended Hardware:

P4 2.5GHz +
2GB RAM
40GB Free Disk Space
T1 Connection

Installation

You need to log in with a user account that has Administrative Privileges to install Web Scraper Plus+. Once it is installed, you can login as a member of the Power Users Group.

Note to Administrators:

You can also run Web Scraper Plus+ as a member of the Users group, but you will receive a dialog when Web Scraper Plus+ attempts to start the database telling you that permission to start the database is denied. Normally, the database is running anyway, so you can probably click past the error without a problem. Running Web Scraper Plus+ as a member of the User local group only is not supported, but if you need to do it for security reasons, it should work.

Tips for Successful Web Automation

- Think of extracting data in terms of two distinct steps:
 1. Downloading the pages locally getting all the pages you want into the same directory.
 2. Extracting data from local files using a datapage editor.

- Use the web crawler as often as you can. It is a simple task to set up and using url rules to filter the links that are crawled it can also be very precise. For example, it can be used very effectively to follow next page links.
- Test your form submit statements in Form Explorer before using them in a package. Automating forms is prone to errors. You will probably need to play around with your form submit statements before they will work every time.
- For the best mix of data performance and convenience, extract your data to the MS SQL instance that comes with Web Scraper Plus+, but connect to the data using Excel. To connect to the SQL database from excel click Data>Import External Data. Choose either New Database Query or Import Data.
- If you need to schedule a package to run at as specific time or interval, use Windows Scheduler to start Web Scraper Plus+ from the command line.

Build a New Web Automation Package

A Web Automation Package is a group of Web Tasks grouped together and executed in a specific order. For Example, suppose you need to do the following:

1. Log in to a secure website.
2. Submit a search form a dozen (or several thousand) times.
3. Crawl, download, and categorize the search results.
4. Extract data from the search results and put it into Excel.

This whole process can be automated using a Web Automation Package and would consist of four Web Tasks.

All Web Tasks have two basic questions that you will need to answer:

1. Where are you getting the web pages from?

Download Task: Answer with a list of Urls, files, or folders.

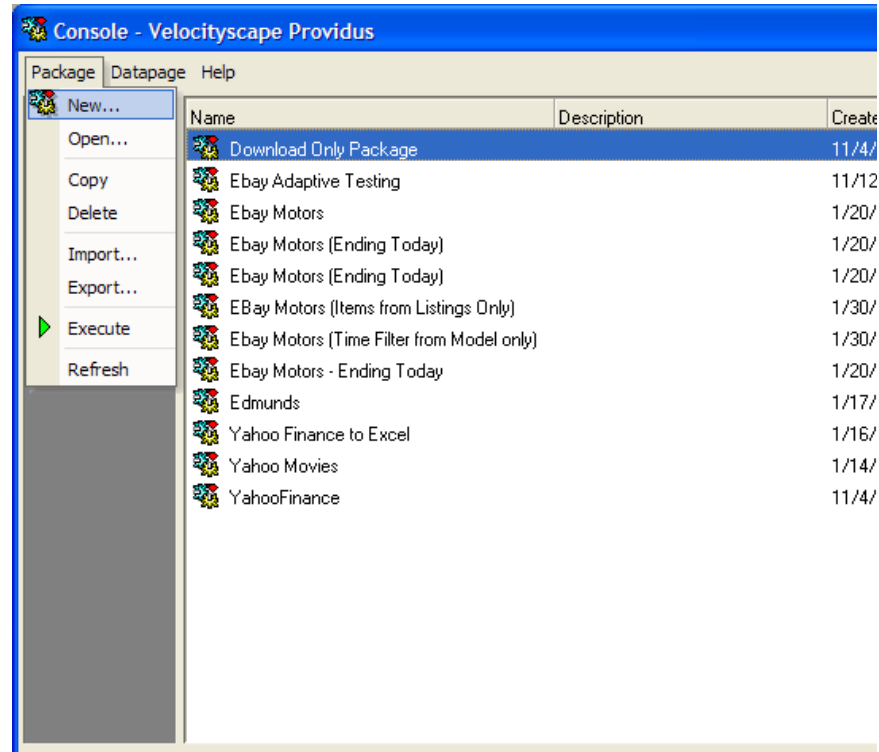
Web Crawler Task: Answer with a Url or local file to start crawling from, and constraints to tell it when to stop.

Form Task: Answer with a list of Form Submit Statements.

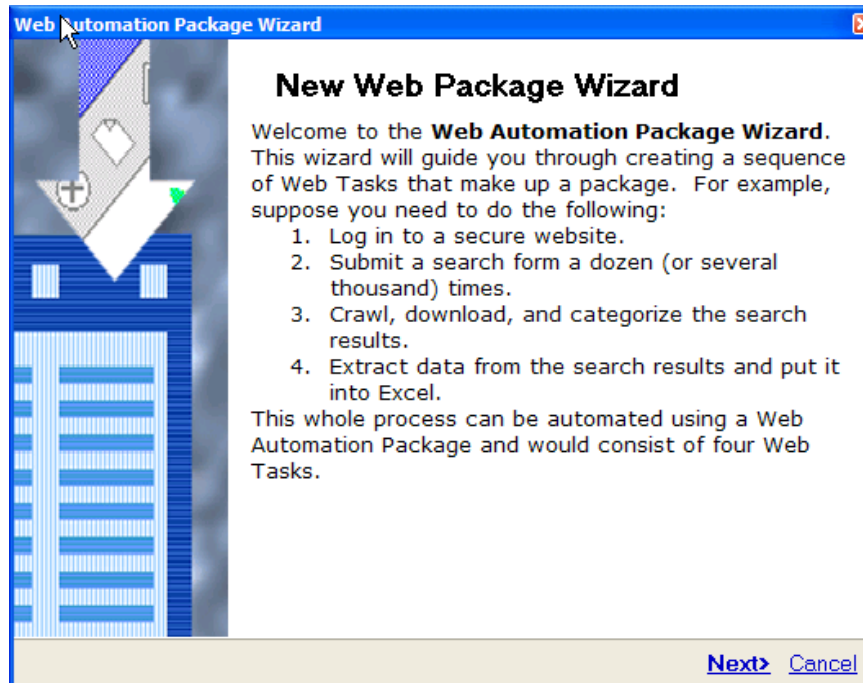
2. What do you want to do with the pages once you have them?

Extract data with a Datapage: In many cases, you will want to extract data from the web pages that you have downloaded and put the data into a database or spreadsheet. To do this, you can build a custom datapage extraction template. A datapage allows you to map fields and sections of a webpage to columns and rows in a spreadsheet or database.

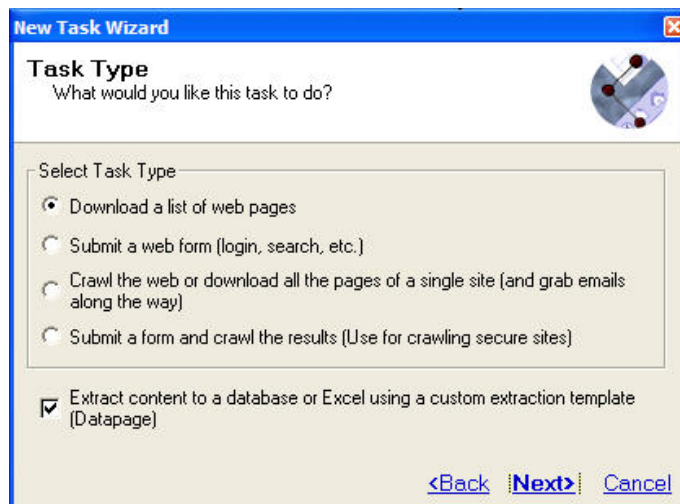
Organize the pages: In some cases, you do not need to do anything with the web pages once you download them. For example, sometimes it is good enough just to submit a form and download the results. You don't necessarily need to extract data from the result page. However, you may still wish to organize the pages that are downloaded. Using a Save As Template, you can organize the pages by Date/Time info, Url parts, Package/Task info, and more.



To create a new package, select package > New from the console.



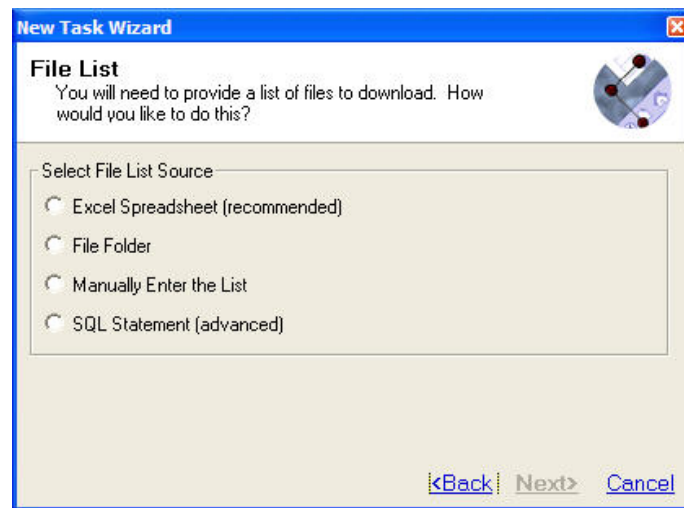
Choose the Type of Task to Create



Choose type of web task that you would like to create based on what you need to do. If you choose Submit form or Download List, then you can also select Extract content using a datapage. Sometimes it makes sense to do a

Download Task

A download task is a list of files to be downloaded (and potentially processed by datapage extraction template). The verb download may be a confusing description, because a download task does not necessarily have to be from some location on the web. A download task can use a combination of windows file paths, windows folders (including subfolders), or internet urls as a source for its file list. This is important in the context of packages, because a web crawler does a good job of downloading and organizing files, but it cannot link directly to a datapage extraction template. So, a web crawler is often used to download an entire website and categorize it by file name, but then a download task is used to extract data to a spreadsheet or database.

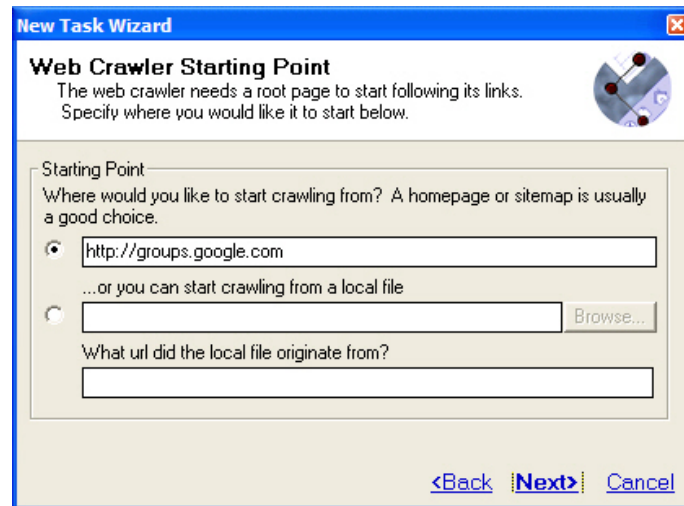


Download tasks can build their file lists in four-task based ways through the Web Task Wizard, and are then edited in two more general technical ways through the Task Properties Form. The task-based ways are: Excel File List, Manual File or Folder List, Files from Folder, and File or Folder List from SQL Statement. The corresponding technical types are Manual List and SQL Statement. The Excel File List generates a datalink and SQL statement to retrieve the data from the spreadsheet.

Web Crawler Task

Web Crawler is a term you are probably familiar with. Also, known as a web spider, it is what search engines such as Google and others use to go out and get the raw

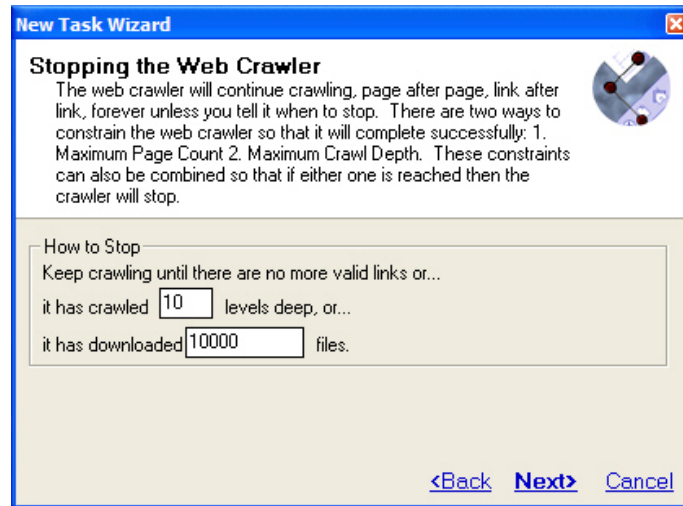
files that they then use to build their indexes. Crawlers work by following links and downloading the web page at the resulting url. Web Scraper Plus+ includes a powerful industrial grade web crawler capable of downloading 200,000+ unique pages per day. Web Scraper Plus+ can also be used to crawl secure websites and search results, something that none of the major search engines are capable of.



Part of what makes the Web Scraper Plus+ web crawler so powerful is its ability to filter which urls to crawl and then its ability to sort the resulting pages as they are downloaded. As an example, you may want to crawl only one site (or perhaps a segment of one site). To do this, you could use url filters to restrict the crawler only to crawl links where the host name includes the base domain that you want to crawl, "buy.com" for example. You may then want to put all of the pages that have the same base file name, "category.asp" for example, into the same directory. That way you could run a datapage extraction template to extract the entire Buy.com product catalog without ever having to teach Web Scraper Plus+ how to navigate the website. The web crawler does all the downloading for you. To see this anecdote in action check out the web package entitled "Buy.com Product Catalog Extraction". For more on url filtering and file sorting, see the corresponding sections below.

Stopping the Web Crawler

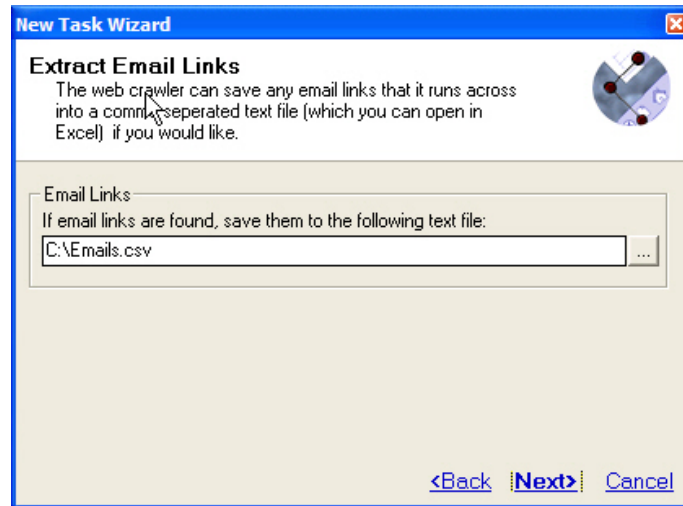
The web crawler will continue to crawl, page after page, link after link, forever unless you tell it how to stop. When a web crawler crawls without bounds, it is called falling into the black hole. If you use url rules to constrain the crawler to download files from a single website, for example, the crawler will stop only when it has downloaded all the pages on that website. But, if you tell it to start at a website and don't use url rules to constrain it to that website, it will crawl to any site that the original site that linked to, and then on to any site that that site linked to and so on until it crawls the entire internet or fills up your hard disk. Of course, if this does happen to you, just hit the cancel button, and all will be fine.



What you can do, either as a safety net, or as a means of constraining an otherwise unbounded web crawler is to set a maximum file count or a maximum crawl depth constraint (you can also set both).

Crawling Emails

In addition to crawling web pages, Web Scraper Plus+ web crawler can also extract emails. In fact, all you have to do to start extracting emails is tell Web Scraper Plus+ where to put the email addresses it finds. While we do not advocate the use of Web Scraper Plus+ for the purpose of building spam lists, it is often useful to crawl a site that you own or are a member of for the purpose of building a targeted list perhaps for sending newsletters.

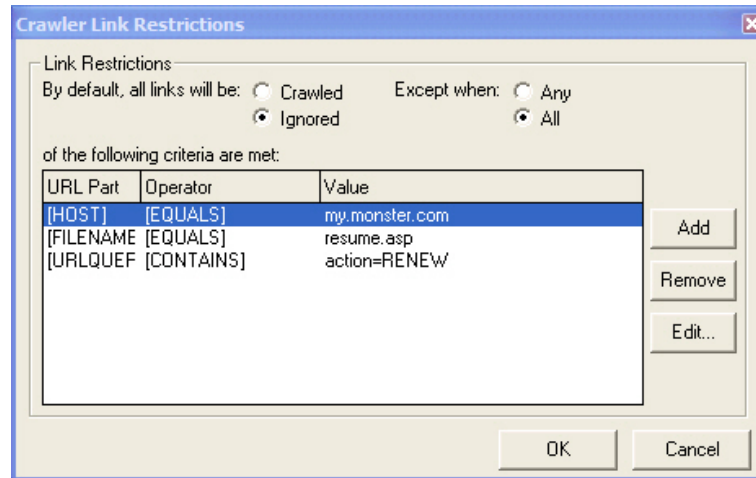


The resulting list of email addresses is comma separated and carriage return + line feed (Enter) terminated. It includes not only the email address, but also the source url from which it was extracted. The file can be opened in Microsoft Excel, but in order to get it into individual columns, you will probably need to select the A Column, then select Data>Text to Columns... In the wizard, select delimited, comma, and set text qualifier to 'none'. The wizard may vary slightly depending on the version of Excel that you use, but in general, that's what you need to do.

Filtering the Links that the Web Crawler Downloads using Url Rules

One of the most powerful aspects of the Web Scraper Plus+ web crawler is its ability to filter the urls that it crawls using a highly customizable set of rules. You can create as many rules as you like, but each rule has three basic parts:

1. The part of the url (if not the whole thing) that you want to evaluate.
2. The value to compare with the url or url part.
3. The operator by which to compare the url part against the comparison value.



For example, you could create a rule that checks to see if a link points the website you are extracting using a rule like “Host Contains ‘yahoo.com’”. In addition to creating the rules, you can also decide how the rules filter the results. You choose whether to crawl links that are matches or to ignore them. So, for our example above, that rule could either be applied to ensure that only pages from yahoo.com are crawled or that no pages from yahoo.com are crawled. Finally, when you have multiple rules, you can choose whether a link must meet ALL the filters or ANY of the filters. This is basically the difference between a logical OR or a logical AND of all the rules in a set.

Parts of a Uri

If you take a typical url like

<http://www.buy.com/retail/computers/category.asp?loc=17078>, you can break it down into several component parts. The Web Crawler allows you to compare seven different url parts when filtering urls:

- Entire Url: There is of course the entire url. In this case that is <http://www.buy.com/retail/computers/category.asp?loc=17078>
- Scheme: In this case the scheme is “http”. In other cases it could be https, or file. The web crawler will not crawl ftp sites, so there is no need to exclude ftp (or telnet, gopher, mailto, etc for that matter).
- Host: In this case the host is www.buy.com. It could be an IP address like 169.42.224.21, but a host of some form will always exist. Host is probably the most commonly used url part in filters because you can use it to ensure that you don’t crawl off of a specific site.
- Path: In this case the path is “/retail/computers/”. Path is a useful if you want to crawl part of a website but not the entire thing. Most websites are organized logically by path, as to an extent is our example. Perhaps a better example is a site like yahoo.com where each directory could be a site in and of itself.

- **Filename:** In this case the filename is “category.asp”. Often, when you want to extract data from a website, you only want to pull data from one templated server side page. That is, the data on the page changes depending on the search criteria, querystring, or something else, but layout of the page does not change. Datapage extraction templates exploit this fact to scrape data. Okay, so when you first start thinking about how to crawl a website, you might think, “Why don’t I just filter all urls except pages that are named the page that I want to extract?” In some cases that may work, but only if your starting page contains a link to that page and all of the data that you want to pull down can be crawled without touching a page without any other name. That works pretty well when your starting page is a set of results and all you want to do is follow the next page link. In the case of Buy.com, for example, it won’t work. You have to crawl the entire site to ensure that you get all the category.asp pages that you need to extract the entire product catalog.
- **Querystring:** In this case, the querystring is “loc=17078”. Websites pass all kinds of interesting data in querystrings. For example, if you want to crawl a specific newsgroup from groups.google.com, you could do so by filtering on the querystring because Google’s urls look like so: <http://groups.google.com/groups?hl=en&lr=&group=biz.comp> See how it works? If you want to stay within the newsgroup biz.comp, just look for “group=biz.comp” in the querystring.
- **File Extension:** In this case, the file extension is “asp”. The web crawler just crawls links and downloads whatever is at the end of them. It will download pdfs, exes, docs, mp3s, etc. If there is a file extension that you do not want to download, just filter them out.

Comparison Operators

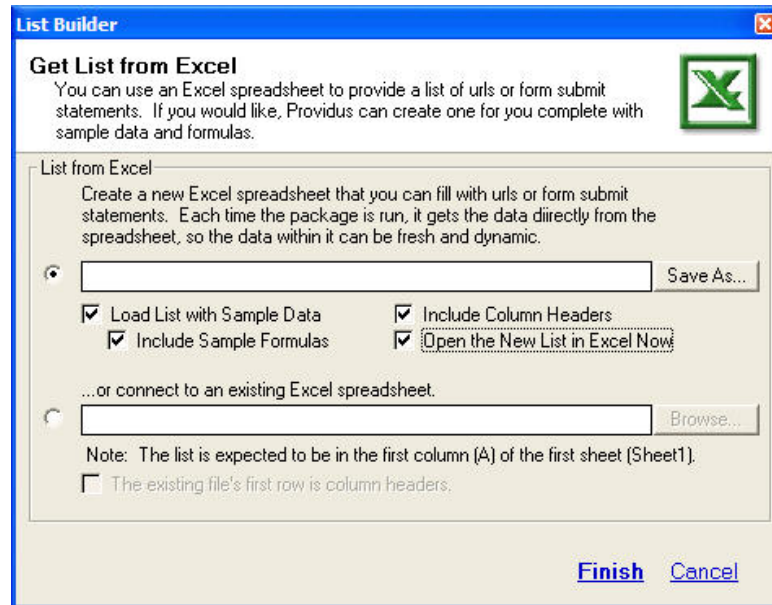
There are 8 operators that you can use to compare a url or url part to a string value. To be technical, there are really only 4 operators and then the logical NOT of those operators. All of the operators compare strings of text as opposed to numbers so the value “05” is not equal to “5”. In addition, the operators are case insensitive, so “AbC” is equal to “abc”. The 8 operators are discussed below:

- **Contains:** Returns true if the url or url part has the string value anywhere within it. For example if the url is <http://www.yahoo.com>, HOST Contains “Yahoo” returns true. Contains is most certainly the most useful operator. It and its logical opposite (Does Not Contain) are the least precise, but most forgiving operators.
- **Does Not Contain:** Returns true if the url or url part does not have the string value anywhere within it. For example if the url is <http://www.yahoo.com>, HOST Does Not Contain “Google” returns true.
- **Equals:** Returns true only if the url or url part and the string value are exactly the same (case insensitive). For example, if the url is <http://www.yahoo.com/biz/index.htm> then PATH Equals “/biz/” returns true.

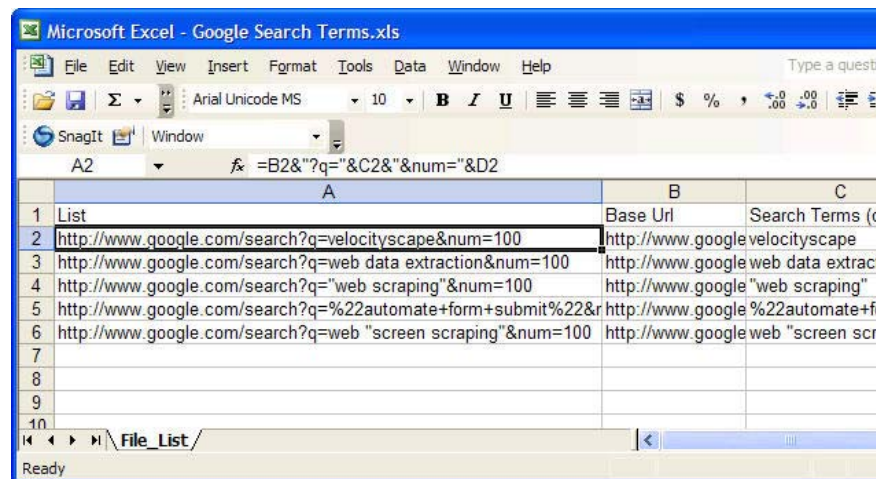
- Does Not Equal: Returns true only if the url or url part and the string value are not exactly the same. For example, if the url is <http://www.yahoo.com/biz/index.htm> then PATH Does Not Equal “/biz” returns true. Did you see the difference? It’s only a slash. Equals and Not Equals are the most precise operators, but also the least forgiving. If you are wrong with an Equals operator, even by a little, you are still wrong.
- Starts With: Returns true if a url or url part starts with the string value. Starts With and Ends With are not recommended for use with the querystring. This is because querystring parameters are not position sensitive. For example, the url <http://www.foo.com?id=abc&pg=10> means exactly the same the thing to a web server as <http://www.foo.com?pg=10&id=abc> . Contrast that with other parts of the url that are case sensitive such as path; <http://www.foo.com/id/pg/> means something completely different to a web server than <http://www.foo.com/pg/id/>.
- Does Not Start With: Returns true if a url or url part does not start with the string value.
- Ends With: Returns true if a url or url part ends with the string value. Starts With and Ends With are not recommended for use with the querystring. This is because querystring parameters are not position sensitive. For example, the url <http://www.foo.com?id=abc&pg=10> means exactly the same the thing to a web server as <http://www.foo.com?pg=10&id=abc> . Contrast that with other parts of the url that are case sensitive such as path; <http://www.foo.com/id/pg/> means something completely different to a web server than <http://www.foo.com/pg/id/>.
- Does Not End With: Returns true if a url or url part does not end with the string value.

Form Submit Task

A Form Task allows you to fill out and submit web forms using a list of form submit statements. You can use a form task to log in to a secure website, submit a search, or anything else you would need to use a form for. When you create a form task, you can have the wizard build you an Excel spreadsheet complete with formulas that generate the form submit statements. Each control on a web form is a column on the spreadsheet. All you have to do is fill in the values.



Create a custom excel template for use in automating forms.

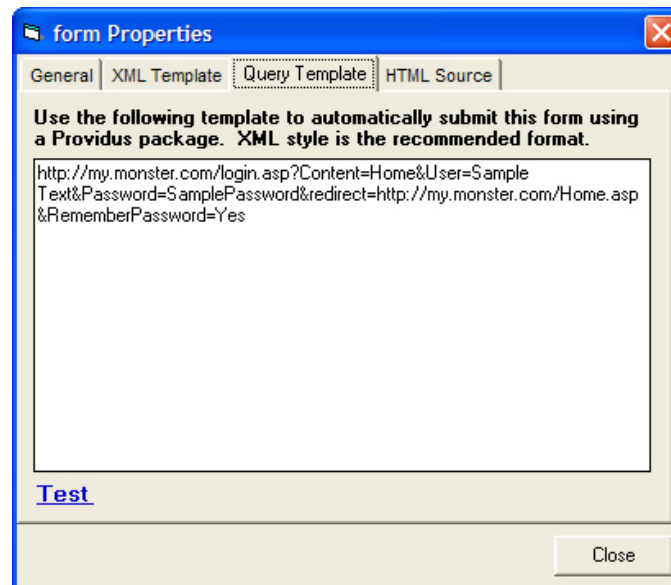


In addition to providing the list of form submit statements from excel, you can also manually enter a list, or you can provide it from any OLEDB or ODBC data source using a simple SQL Select statement.

If you've never heard of a form submit statement before, don't worry, you're not losing your edge. We invented it. Basically, a form submit statement is a simple way of telling Web Scraper Plus+ what form and what values to submit. A form submit statement can be expressed in two formats: querystring style and xml style. A custom template for each of these formats can be generated by the Task Wizard and the Form Explorer from any web form, so what follows is informative enough, but you will not be tested on it later.

Querystring Style Submit Statement (Simple but Limited):

Querystring style submit statements look exactly like a standard url with a querystring. For example, if you had a Google Adwords account and your username was "foo@bar.com" and your password was "myPass", then you could login to your account using the following querystring style submit statement:
<https://adwords.google.com/select/LoginValidation?login.userid=foo@bar.com&login.password=myPass>



The process of building a querystring formatted submit statement from scratch is simple: Form Action Url (<http://adwords.google.com/select/LoginValidation>) + Question Mark (?) + First Control Id (login.userid) + Equals (=) + Control Value (foo@bar.com) + Ampersand (&) + Next Control Id (login.password), and so on. Of course, you will never need to create a submit statement from scratch because both the Task Wizard and the Form Explorer will create templates for you for any form whenever you need one.

The querystring submit statement is handy and compact, but it will not work in all circumstances. Sometimes a form's action url has a querystring already. Such is the case with the EBay Login form:

https://signin.ebay.com/ws/eBayISAPI.dll?co_partnerid=2&siteid=0&UsingSSL=1 In this case, you

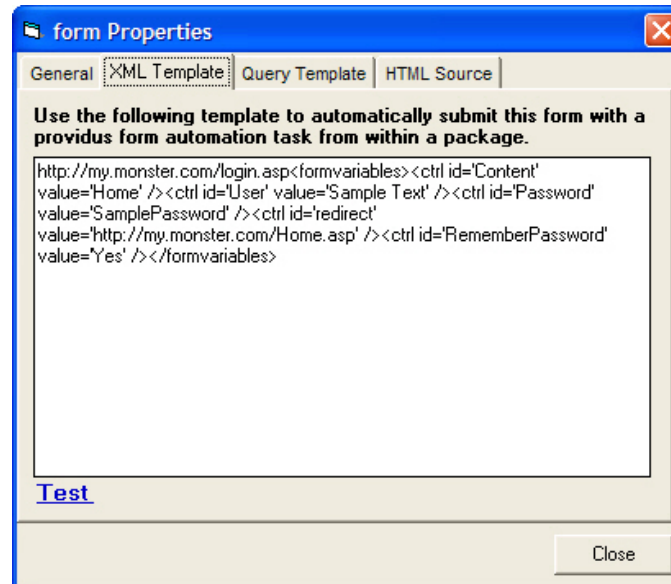
should not use the querystring format because the existing querystring parameters (co_partnerid, siteid, UsingSSL) would also be submitted as form values.

The querystring style submit statement also has other limitations. It does not support Tabs, Carriage Returns, or Line Feeds. It also does not support the following special characters (even escaped): "=", "&", "%", "+", "?", "#". So, the querystring format is handy, but it has its limits, and in all but the most basic cases (logins, simple searches, etc.) it is not encouraged.

XML Style Submit Statement (A Little More Complicated, But Worth it):

XML style submit statements look like HTML or XML, hence the name. You don't need to know anything about XML or HTML to understand it though. For example, if you had a Google Adwords account and your username was "foo@bar.com" and your password was "myPass", then you could login to your account using the following XML style submit statement:

```
https://adwords.google.com/select/LoginValidation<formvariables><ctrl id='login.userid' value='foo@bar.com' /><ctrl id='login.password' value='myPass' /></formvariables>
```



The process of constructing a XML style submit statement from scratch is pretty simple: Form Action Url (<http://adwords.google.com/select/LoginValidation>) + Form Variables Tag ("`<formvariables>`") + First Control Tag (`<ctrl id=`) + First Control Id ("`login.userid`") + "value=" + Control Value (foo@bar.com) + End the First Control Tag ("`>`") + Next Control Tag ("`<ctrl id=`"), and so on... + End of Form Variables (`</formvariables>`). Of course, you will never need to create a submit statement from scratch because both the Task Wizard and the Form Explorer will create templates for you for any form whenever you need one.

The most important thing to remember about XML style submit statements:

The single quotes (') that surround the control id and value are absolutely required! If you do not include them, then you will get an error.

Other things you might forget:

Don't forget to include the "<<formvariables>>" and "<</formvariables>>" tags around the control tags.

Don't forget to include an end to the ctrl tag ("</>").

The XML style submit statement tags and attributes (<formvariables>, <ctrl>, id, name) are **NOT case sensitive**. There are also some aliases that also work such that:

<formvariables> = <FormVars> = <Form_Ctrls> = <FormControls>

<ctrl> = <variable> = <Var>

Id =name

Value = val

Tips on Form Automation

- When a template form submit statement is created, it includes sample values for all of the controls that do not already have a value. So, a Hidden control may not have any value at all in the HTML of the form, but the template generates a value like 'Sample Hidden Value'. You should change the value from 'Sample Hidden Value' to either empty '' or something meaningful. If you do not know of something meaningful, just set it to empty.
- When a template form submit statement is created, it includes the name and values of all buttons on the form. When there is more than one submit button on a form, you will want to delete all but one of the submit buttons. Consider the case of Google.com. There are two submit buttons on their basic search form: "Google Search" and "I'm Feeling Lucky". If you include both of their controls in your submit statement Google will redirect you to the first url in the results (as if you clicked I'm Feeling Lucky"). You need to remove the second button control all together if you want the search to run as normal.
- When a form task is created, by default the referrer url of the http client is set to the page that the form is on. That is usually the right value for the task that submits the form. But, since future tasks inherit all http client values that are not explicitly changed, you may need to set the referrer url to something different in the next task. If you leave it alone and let referrer be inherited, in most cases nothing changes, but in some cases you will get mixed, misleading, or error pages.

Testing Form Submit Statements

Automating forms is prone to errors. You will probably need to play around with your form submit statements before they will work every time. You can test your form submit statements from the form explorer. Simply navigate to the form that you want to automate. Double-Click on the Form node in the tree pane to bring up form properties. Click on either the XML or Query tab and click the test button in the lower left corner.

Form Submission Testing

Test Form Submit Statement

Edit the form submit statement in the text window below. Click 'Test' to launch the result of the statement in a new browser window. The test functions as if it is in the first task of a package, so if you are currently logged into a secure site, it will not take that into account. In that case, the only way to test is to run the package and examine the results.

```
http://my.monster.com/login.asp<formvariables><ctrl id='Content' value='Home' /><ctrl id='User' value='JSimpson' /><ctrl id='Password' value='HelloKitty' /><ctrl id='redirect' value='http://my.monster.com/Home.asp' /><ctrl id='RememberPassword' value='Yes' /></formvariables>
```

[Advanced](#) [Save Statement](#) [Test](#) [Cancel](#)

Organizing Downloaded Files with a Save as Template

Save As Templates

One of the most powerful features of Web Scraper Plus+ is its ability to sort files into different directories and give them different names based on a system of wildcards. For each task, you can define 3 levels of folder hierarchy and a filename. You use all of the levels of the hierarchy or you can use none. If you do not use any of the folder hierarchy, all files that are downloaded will be placed directly into the Root folder. If you don't specify anything for the filename, all of the files will be named default.htm. In addition to using wildcards, you can also use static strings, and a combination of wildcards and static strings. For example, "[TASK_NAME]_Downloaded_From_[HOSTNAME]" is a valid value.

The levels of the folder hierarchy are named Package Folder, Task Folder, and Document Folder, which are suggestions as to how they can be used. In fact, the folders are created dynamically based on the value of any wildcard or static value at the time that the file is saved, so there is no specific link to a Task or Package. All folders will be created if they do not already exist as subfolders of the Root Folder.

Valid Save As wildcards are as follows:

Url wildcards: [URL], [URL_PATH], [URL_FILENAME], [HOSTNAME], and [QUERY_STRING]

Date wildcards: [DATE] or [TODAY], [DAY] or DD/ or MM- or /DD or -DD, [MONTH] or MM/ or MM- or /MM or -MM, [YEAR] or /YYYY or -YYYY or YYYY/ or -YYYY, DD-MM-YYYY

Time wildcards: [TIME], [DATETIME], [HOUR] or HH:, [MINUTE] or :MM, [SECOND] or :SS, and MM/DD/YYYY_HH:MM:SS

Package wildcards: [PACKAGE_NAME], [PACKAGE_TIME], and [PACKAGE_ID]

Task wildcards: [TASK_NAME], [TASK_TIME], and [PACKAGE_ID]

Other wildcards: [DATAPAGE_NAME], [DOCUMENT_ID]

The Package, Task, and Other wildcards are useful for providing a system of separating the results of one package or task from another. The Date and Time wildcards are useful for versioning the results of the instance of the package that you

ran an hour or week ago from the one you are running now. Perhaps the most interesting wildcards are the Url wildcards.

Using the Url wildcards you can group files by host, path, filename, and even querystring. If you want to extract data that comes from a dynamic page like ContactList.asp using a datapage, you can isolate all of the files named ContactList.asp into as single directory by using the [URL_FILENAME] wildcard in the "Document Folder". If you want to crawl several sites and be able to distinguish where the results came from you could put [HOSTNAME] in the Task Folder. If you want a hierarchy of local folders to look the same as the folder structure of the website, put [URL_PATH] in the Task Folder. We are constantly coming up with new and interesting ways to use these wildcards to make what we want to get out of the data easier to get. Our best advice is to experiment with wildcards, because while they seem obvious at first, their power to make you life easier grows with experience.

Examples:

For all the samples below, assume the following task and package properties:

Package and Task Properties

Package Name: "Test Package"

Package Start: 10/22/2004 3:14 PM

Task Name: "Task1"

Example 1 – Basic Versioning

Save As Template Properties

Root Folder: C:\Downloads\

Package Folder: [PACKAGE_NAME]_[PACKAGE_DATE]

Task Folder: [TASK_NAME]

Document Folder: <empty>

Filename: [URL_FILENAME]_[QUERY_STRING]

Url: <http://employers.monster.com/resume.asp?id=18802013>

Save to:

C:\Downloads\Test_Package_10_22_04\Task1\resume.asp_id=18802013.htm

Example 2 – Isolate Files

Save As Template Properties

Root Folder: C:\Downloads\

Package Folder: [HOSTNAME]

Task Folder: <empty>

Document Folder: [URL_FILENAME]

Filename: [URL_FILENAME]_[QUERY_STRING]

Url: <http://www.buy.com/retail/category.asp?dept=224>

Save to: C:\Downloads\www_buy_com\categories.asp\category.asp?dept=224.htm

Url: <http://www.buy.com/retail/product.asp?sku=22910928>

Save to: C:\Downloads\www_buy_com\product.asp\product.asp?sku=22910928.htm

Example 3 – Mimic Server Hierarchy

Save As Template Properties

Root Folder: C:\Hoovers_com\

Package Folder: <empty>

Task Folder: [URL_PATH]

Document Folder: <empty>

Filename: [URL_FILENAME]

Url: <http://hoovers.com/company/executives/detail/12319941.xhtml>

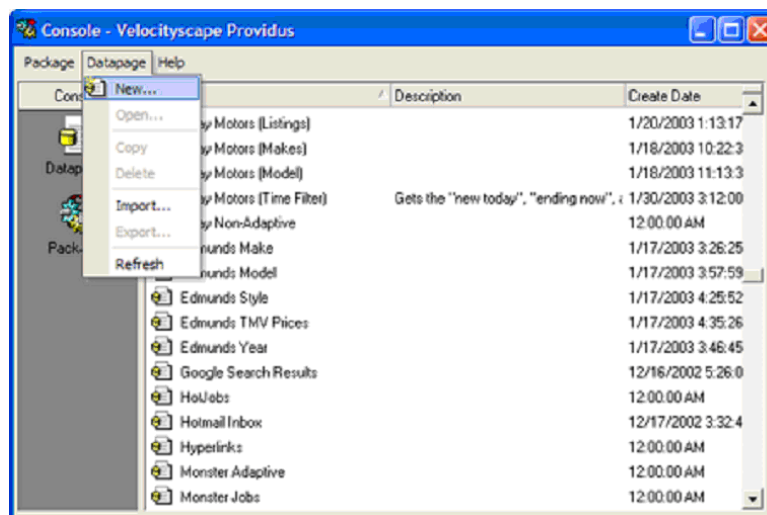
Save to: C:\Hoovers_com\company\executives\detail\12319941.htm

Create a New Datapage

Before you can create a datapage, you should probably know what one is. A datapage is the root of an extraction template. As its name suggests, it maps directly to a webpage. A datapage consists of one or more datasets. Datapages can be linked together to "walk" entire site hierarchies through the use of packages.

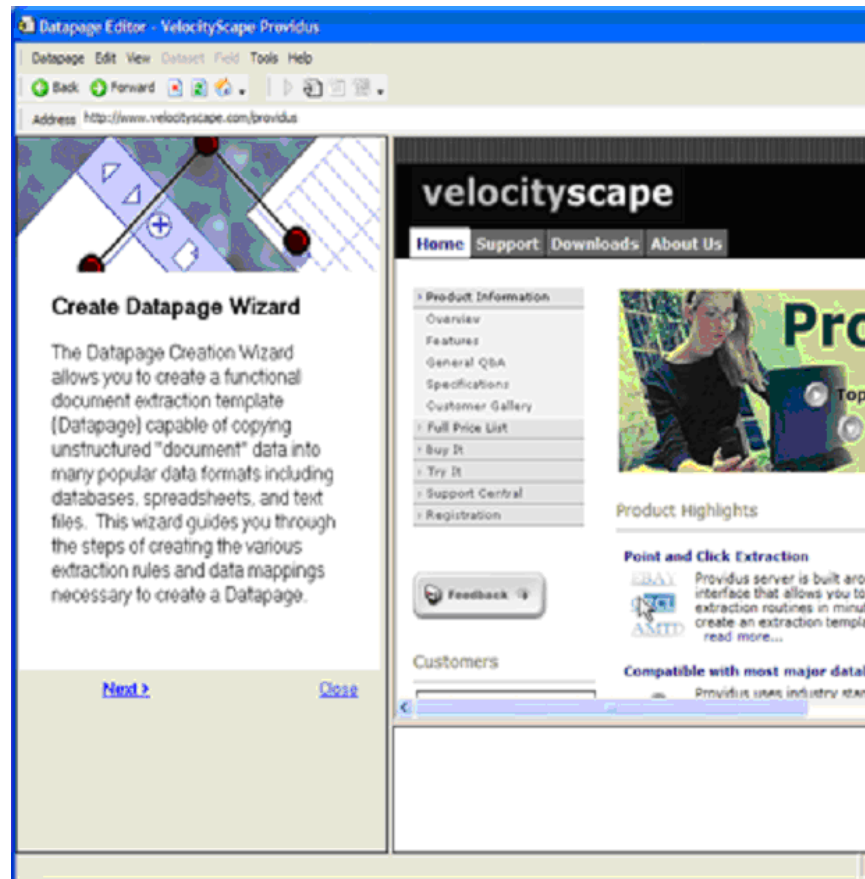
The following is a step by step tutorial of how to create a datapage using the New Datapage Wizard.

Launch the Datapage Editor from the Web Scraper Plus+ Console



Select Datapage > New or Right Click and select New Datapage.

Launch the New Datapage Wizard



The Create Datapage Wizard will automatically launch when the Datapage Editor Starts. Click Next> to continue.

Navigate to the Template Web Page

The screenshot shows the 'Datapage Editor - VelocityScape Providus' interface. On the left, a tutorial titled 'Open Document to Use as Template' explains that users should navigate to a webpage in the 'Browser Pane' to use as a template for data extraction. It includes a tip to choose a page with dynamic data and instructions to click 'Next' to continue. On the right, the 'Yahoo! Finance' page for CSCO is displayed, showing the date 'Wednesday, November 13 2002 7:19pm ET - U.S. Mar', a 'Welcome' message with a 'Sign In' link, and a 'Quotes' section with a search bar and a table of stock data.

Symbol	Last Trade	Change	
CSCO	4:03pm	13.42 +0.55	+4.27%

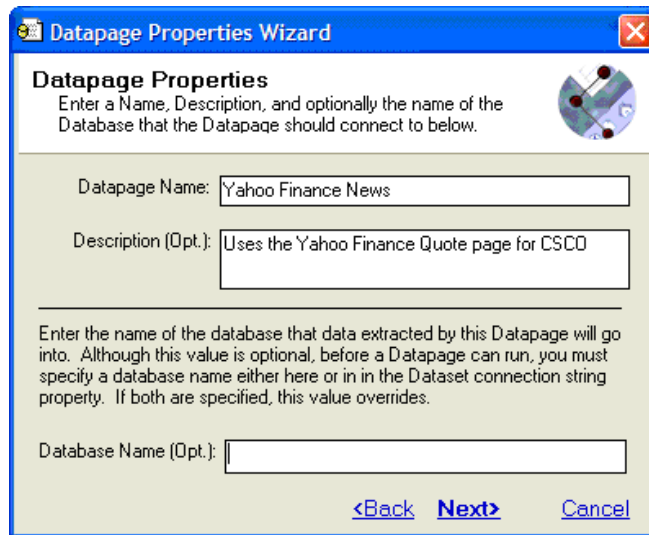
Navigate the Browser pane to the web pages that you wish to use as a template for the extraction of other web pages. Choose this page carefully. This is the foundation of your datapage extraction template. You should make sure that it contains as many permutations and inconsistencies as possible. This will make it so that the test extractions that you will perform later will best stress your datapage. Click Next> to Continue.

Launch the Datapage Properties Wizard



The Datapage Properties Wizard is launched automatically. Leave the (Advanced) checkbox unchecked if you wish to use this wizard in the future. If you check it then the standard Datapage Properties dialog will be launched instead. Click Next> to continue.

Specify the Base Datapage Attributes



The screenshot shows a Windows-style dialog box titled "Datapage Properties Wizard". The main heading is "Datapage Properties". Below the heading is a sub-heading "Enter a Name, Description, and optionally the name of the Database that the Datapage should connect to below." followed by a small globe icon. There are three text input fields: "Datapage Name:" with the value "Yahoo Finance News", "Description (Opt.):" with the value "Uses the Yahoo Finance Quote page for CSCO", and "Database Name (Opt.):" which is empty. At the bottom right, there are three buttons: "<Back", "Next>", and "Cancel".

Enter a Name and Description for the New Datapage and optionally the name of the database that the datapage will be extracted to. The Database name field is optional here, but a database name must be specified either here or at the dataset level. If it is specified here and at the dataset level, this value takes precedent. Click Next> to continue, then Click Finish to complete the Datapage Properties Wizard.

Create a New Dataset

The next step in the creation of the New Datapage Wizard is to create a Dataset. A dataset exists within a datapage. It is directly analogous to a table (or sheet). It is made up of fields that map to columns. As an example, think of a listing of items on EBay. The entire list is a dataset. Each individual item is a Datarow and each attribute (Current Price, Title, etc.) is a field.

Select a Dataset

Select the Beginning of the Dataset

What set of data in the web page or document do you wish to extract? Create a template for the dataset by marking the beginning of one of the "rows" of the dataset.

In the **Browser Pane**, select one of the data items (rows) in your dataset as a template for extraction. Choose one that accurately represents the diversity of data in the dataset.

Select a template Data Item (row) in the Browser Pane

Wed 9:02am CSCO [external] Biogen up on upgrade, Sobel System [external] at CBS MarketWatch

Wed 8:11am CSCO [SS - free trial] Rally's and the Personal Peter Lynch Effect - RealCommentary from TheStreet.com

Tip: For best results, select the entire template data item(row) and the beginning of the next item. This should ensure that all of the underlying source is selected.

Click Next to continue.

[< Back](#) [Next >](#) [Advanced](#) [Close](#)

Recent News

New research reports for [CSCO](#)

Wed 2:52pm CSCO [Riverstone to supply routers to Korea's Reuters](#)

Wed 2:18pm CSCO [\[SS - free trial\] Cisco Falls, Then Rises o News - RealCommentary from TheStreet](#)

Wed 9:10am CSCO [\[external\] A 99-Cent Hamburger Isn't Def TheStreet.com](#)

Wed 9:09am CSCO [\[external\] MARKET MOVERS: TRC Tur Missing Estimates - at BusinessWeek C](#)

Wed 9:03am CSCO [\[SS - free trial\] Rally's on its Last Legs - RealCommentary from TheStreet.com](#)

Wed 8:07am CSCO [Tokyo Stocks Drop to 19-Year Lows - A Press](#)

Tue 10:15pm CSCO [Tokyo Stocks Fall 54 Points - Associate](#)

Tue 7:31pm CSCO [Chief Execs Sour on Next Year's Outloo](#)

Tue 6:35pm CSCO [UPDATE 1-After the Bell-Nordstrom slur Network Appliance up - Reuters](#)

Tue 6:30pm CSCO [\[SS - free trial\] Faith in Tech Trumps Tra RealCommentary from TheStreet.com](#)

```
<TR vAlign=top><TD noWrap><FONT face=arial size=1>Wed </FONT></TD>
<td colspan=2><FONT face=arial size=1>2:52pm </FONT></TD><FONT fac
</TD><TD><FONT face=arial size=1><A href="http://biz.yahoo.com/nc/02111
1.html">Riverstone to supply routers to Korea's Hangoo</A> - Reuters</FONT>
<TD noWrap><FONT face=arial size=1>Wed </FONT></TD></TR>
```

Select one of the datarows in the dataset. In this case, Recent News is the dataset and each news article is a datarow. You can select any of the datarows, but try to select one that includes all optional fields. As an example of this, imagine eBay where there are 50 item listings (datarows) on a page, but only some have a Buy it Now price. In such a case, you would want to select one of the datarows that has a Buy It Now price. In the case illustrated above, there are not optional columns, so we can just select the first row. But note that the beginning of the second row was also selected. This helps to ensure that all of the underlying source (displayed in the lower selection pane) is included in the selection. Click Next> to Continue.

Select the Number of Datarows in the Dataset

Specify the Number of Items (Rows) in the Dataset
How many times does the data item appear in the Datapage? Select from a list of choices below.

The list below contains a complete list of the possible number of data items (rows) in the datapage. Count the number of data items on the datapage in the [Browser Pane](#) and select that number from the drop down list below.

Note: If the number of times the data item appears on the datapage is not in the list below, select 'None of the Above' from the list, and click Next. You will need to manually select the first character of the template data item(row). If after several attempts the correct count still does not appear, select the value closest to your count.

Number of Data Items on the Datapage:

167
37
18
12
10
6
2
1

Continue

Close

Recent News

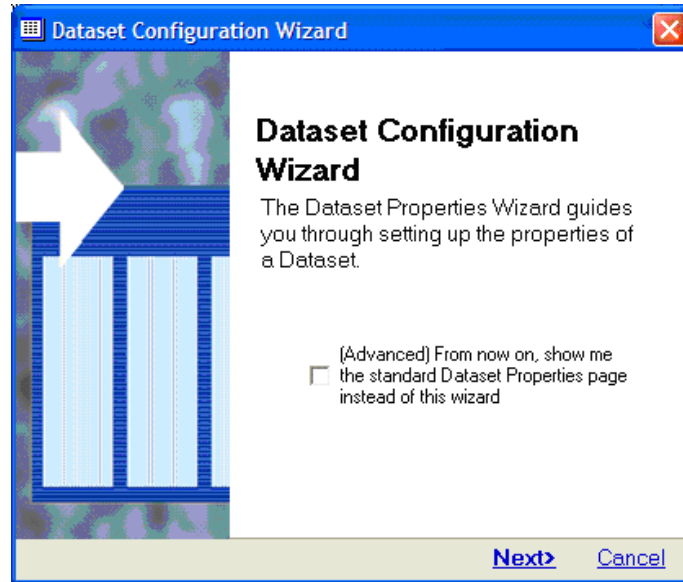
New research reports for CSCO

- Wed 2:52pm CSCO [Riverstone to supply routers to Korea's Hanaro](#) - Reuters
- Wed 2:18pm CSCO [\[SS - free trial\] Cisco Falls, Then Rises on News](#) - RealCommentary from TheStreet
- Wed 9:10am CSCO [\[external\] A 99-Cent Hamburger Isn't Del](#) - TheStreet.com
- Wed 9:09am CSCO [\[external\] MARKET MOVERS: TRC Turn](#) - Missing Estimates - at BusinessWeek C
- Wed 9:03am CSCO [\[SS - free trial\] Rally's on its Last Legs](#) - RealCommentary from TheStreet.com
- Wed 8:07am CSCO [Tokyo Stocks Drop to 19-Year Lows](#) - AP Press
- Tue 10:15pm CSCO [Tokyo Stocks Fall 54 Points](#) - Associate
- Tue 7:31pm CSCO [Chief Execs Sour on Next Year's Outlook](#)
- Tue 6:35pm CSCO [UPDATE 1-After the Bell-Nordstrom slur](#) - Network Appliance up - Reuters
- Tue 6:30pm CSCO [\[SS - free trial\] Faith in Tech Trumps Tr](#) - RealCommentary from TheStreet.com

```
<TR align=top><TD nowrap><FONT face=arial size=1>Wed </FONT></TD><td colspan=2><FONT face=arial size=1>2:52pm </FONT></TD><FONT face=arial size=1><A href=http://biz.yahoo.com/ic/021111.html>Riverstone to supply routers to Korea's Hanaro </A> - Reuters </FONT></TD nowrap><FONT face=arial size=1>Wed </FONT></TD></TR>
```

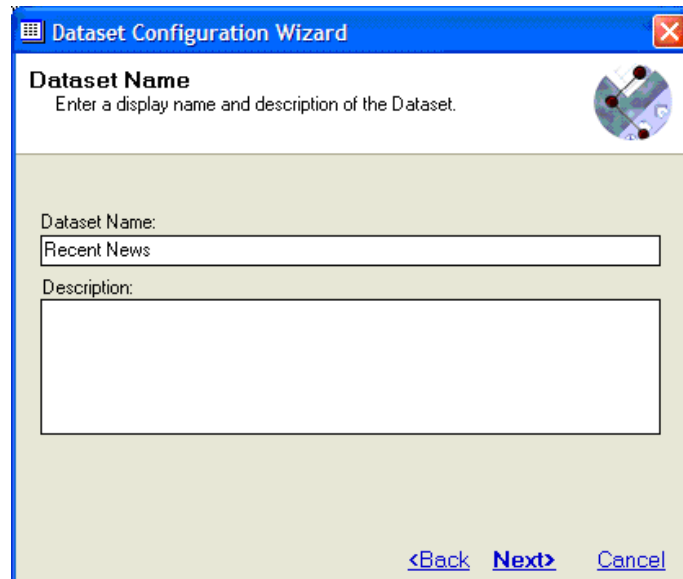
Based on the template row you selected in the last step, Web Scraper Plus+ will calculate every possible datarow count in the dataset based on the underlying source of the web page. Count the number of datarows on the datapage and select that number from the drop down list on the Wizard. If the number that you count is not in the list select 'None of the Above'. You will need to manually select the beginning of the underlying source of the template datarow. Click Next> to continue.

Dataset Configuration Wizard



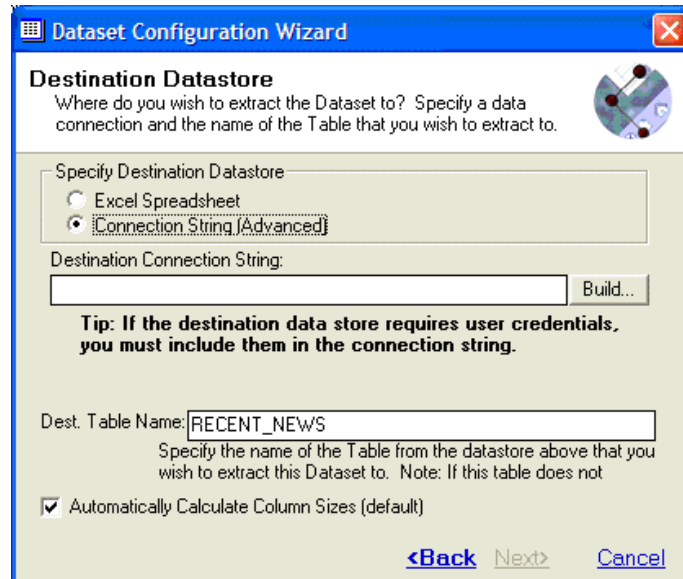
The Dataset Configuration Wizard is launched automatically. Leave the (Advanced) checkbox unchecked if you wish to use this wizard in the future. If you check it then the standard Dataset Properties dialog will be launched instead. Click Next> to continue.

Specify Base Dataset Attributes



Specify the Dataset name and optionally a description. Click Next> to continue.

Specify the Destination Datastore

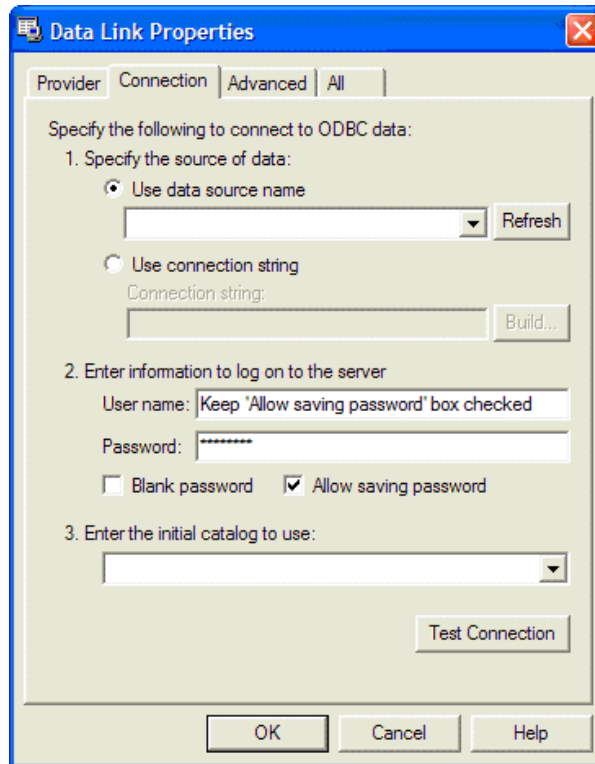


The screenshot shows a dialog box titled "Dataset Configuration Wizard" with a close button (X) in the top right corner. The main heading is "Destination Datastore" with a sub-heading: "Where do you wish to extract the Dataset to? Specify a data connection and the name of the Table that you wish to extract to." Below this, there is a section titled "Specify Destination Datastore" containing two radio button options: "Excel Spreadsheet" and "Connection String (Advanced)". The "Connection String (Advanced)" option is selected. Underneath, there is a text field labeled "Destination Connection String:" followed by a "Build..." button. A tip is displayed: "Tip: If the destination data store requires user credentials, you must include them in the connection string." Below the tip is another text field labeled "Dest. Table Name:" with the value "RECENT_NEWS" entered. A note below the field reads: "Specify the name of the Table from the datastore above that you wish to extract this Dataset to. Note: If this table does not". At the bottom left, there is a checked checkbox labeled "Automatically Calculate Column Sizes (default)". At the bottom right, there are three buttons: "<Back", "Next>", and "Cancel".

Specify the destination datastore for the dataset that you are configuring. You can choose to extract directly to a Microsoft Excel Spreadsheet or you may specify a connection to an OLEDB or ODBC datastore.

If you choose to extract to an Excel spreadsheet, you will can click 'Browse...'' to choose an existing spreadsheet or may enter a new filename (be sure to include the full path including the .xls file extension like "C:\Temp\YahooFinance.xls")

If you choose to extract to an OLEDB or ODBC datastore, you may either enter a known Connection string directly or click the 'Build...'' button. If you do this, the following dialog appears:



This dialog has its own help file, but it is important to note that if you are required to log in to the datastore you wish to extract to (which is most likely the case) you will need to check the “Allow saving password” checkbox. Rest assured that even though this information is stored in the Web Scraper Plus+ database, it is encrypted using 256 bit AES security (higher encryption than most financial institutions currently employ). If you did not specify a database name in the Datapage Properties Wizard, then you must now specify an initial catalog (or similar property depending on your data provider). Once you have set up your connection, click ‘OK’.

Specify the name of the table that this Dataset corresponds to. Remember that there is a one-to-one analogy between a dataset and a database table or an Excel worksheet.

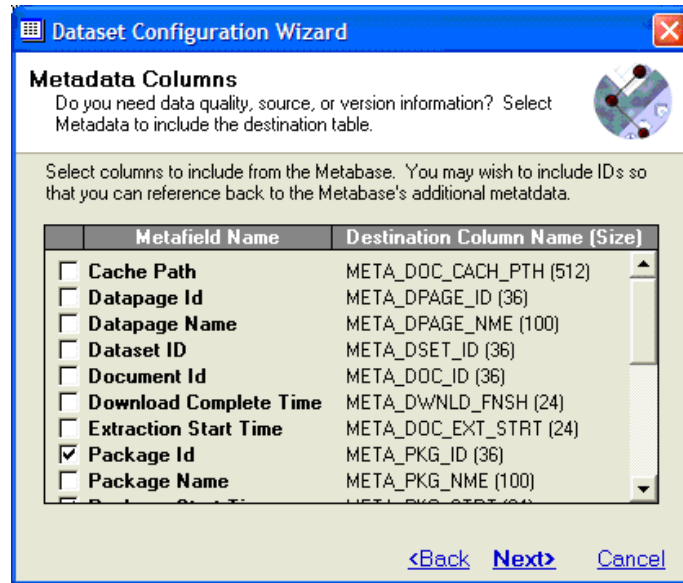
The table (along with its fields) that you specify here will be automatically created on the datastore if the datastore is MS Excel, MS Access, MS SQL Server, or any JET3.51 or 4.0 provider.

Specify whether you want the column size to be automatically calculated for each of the fields that will belong to this dataset. Note that, while convenient and relatively error resistant, this may not be the most efficient way to store your extracted data. Automatic column size calculation uses the largest possible row size for the data store you are using. It splits up the column size between all fields relative to the

template field size. It is recommended that you allow Web Scraper Plus+ to make these calculations initially and optimize for performance later if necessary.

Select Metadata

Web Scraper Plus+ integrates data quality and audit trail metadata into every extraction. This means that for each piece of information that you extract, you know where, when, and how it was extracted and how accurate the extraction was. This gives you the ability to filter out of date or unreliable data. It also gives you vital information necessary to improve the quality of future extractions.



Select the Metadata you wish to include the table (specified in the previous step) for this Dataset. A column will be created on the table for each metafield selected using a column name and size of the corresponding 'Destination Column Name (Size)'. Note that all of this information is stored in the Web Scraper Plus+ Metabase, but selecting Metadata Columns here will store a copy of the information in the destination table. Furthermore, even though the information is stored in the Metabase it will not be possible to relate a row of data in the destination table to the Metabase unless the IDs of the type of Metadata you wish to view is stored in that row. It is recommended that at the very least you store the Document Id. You may also find frequent use for Source URI and Cache Path for the purpose of linking datapages. When you have selected the Metadata you wish to include click Next> to continue. Click finish on the Next screen to complete the Dataset Configuration Wizard.

Create a New Field

A dataset is composed of one or more fields that represent an individual attribute of the entity that the dataset represents. So, in our example, there are 6 fields in the Recent News dataset (Day, Time, Symbol, Headline, Source, and ArticleURL). So, the steps below will need to be repeated six times, once for each field.

Select the Field

The screenshot shows the 'Datapage Editor - VelocityScape Providus' application. The main window is titled 'Add New Data Field' and contains the following text:

Select a field from within the template data item.

Creating a new data Field is a two step process:

1. In the **Browser Pane**, select one of the fields that make up the template data item (row). Its underlying source will appear in the **Selection Pane**.
2. In the **Selection Pane**, select the underlying source of the field that you wish to extract.

Below the instructions, there is a 'Template Data Item (Row)' section showing a book titled 'Online Competitive Intelligence: Increase Your Profits Using Cyber-Intelligence' by Helen Burwell, et al. The price is listed as '\$18.17'. A red callout box points to the price with the text: 'Select a Field from the Template Data Item in the Browser Pane'.

Below the selection pane, there is a text box containing the HTML code: '<B class=price>\$18.17/>'. A red callout box points to the price with the text: 'Select the Underlying Source of the Field in the Selection Pane'.

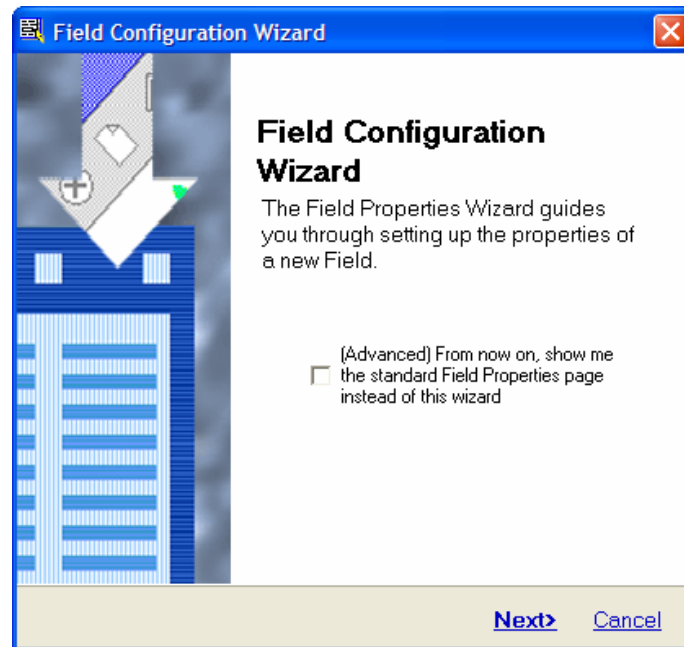
At the bottom of the dialog, there are buttons for '< Back', 'Next >', and 'Close'.

On the right side of the screenshot, there is a 'Recent News' dataset with the following entries:

Day	Time	Symbol	Headline	Source
Wed	2:52pm	CSCO	Riverstone to supply routers to	Reuters
Wed	2:18pm	CSCO	[\$\$ - free trial] Cisco Falls, The	News - RealCommentary from
Wed	9:10am	CSCO	[external] A 99-Cent Hamburge	TheStreet.com
Wed	9:09am	CSCO	[external] MARKET MOVERS:	Missing Estimates - at Busines
Wed	9:03am	CSCO	[\$\$ - free trial] Rally's on Its La	RealCommentary from TheStre
Wed	8:07am	CSCO	Tokyo Stocks Drop to 19-Year	Press
Tue	10:15pm	CSCO	Tokyo Stocks Fall 54 Points -	
Tue	7:31pm	CSCO	Chief Execs Sour on Next Year	
Tue	6:35pm	CSCO	UPDATE 1,After the Bell,Norpe	

When you make a selection in the browser pane, the underlying HTML of the selection appears below in the browser pane. **The selection in the selection pane is the one that matters.** Select the dynamic or content portion of the underlying source in the data row you selected as a template for the dataset. Web Scraper Plus+ will build the extraction template based on the source immediately surrounding the selection in the selection pane. When you have made your selection, click Next> to continue.

Field Configuration Wizard



The Field Configuration Wizard is launched automatically. Leave the (Advanced) checkbox unchecked if you wish to use this wizard in the future. If you check it then the standard Field Properties dialog will be launched instead. Click Next> to continue.

Specify Field Name and Whether it is Optional



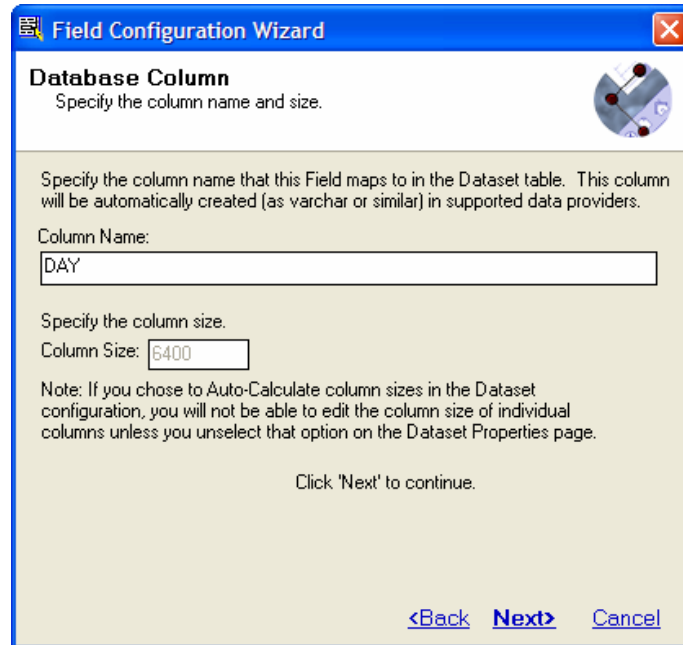
The screenshot shows a dialog box titled "Field Configuration Wizard" with a close button in the top right corner. The main heading is "Field Basics" with a sub-instruction: "Enter a display name and choose whether the field is optional." Below this is a text input field labeled "Display Name:" containing the text "Day". A section titled "Adaptive Extraction" contains a paragraph of text explaining that the wizard can dynamically adapt to optional fields using a proprietary algorithm, and that fields not present in every row should be marked as optional. At the bottom of this section is a checkbox labeled "Optional Field" which is currently unchecked. At the bottom of the dialog box are three buttons: "<Back", "Next>", and "Cancel".

Specify the logical name for the field. The default will be the value of the template field. In most cases there is a more generic name that would apply to all possible values.

Sometimes datasets are not completely predictable. For example, every item on EBay does not have a "Buy It Now" price, but some do. Traditional extraction techniques fail under these circumstances. Web Scraper Plus+ does not. It uses patent pending Heuristically Adaptive Extraction Algorithms (HAXA) that allow you to set data fields as optional. This way if the dataset is not completely homogeneous, the extraction will still return extremely good results. Check the 'Optional Field' checkbox if this field appears in some data rows, but not others.

Click 'Next>' to continue.

Specify Column Name and Size



The screenshot shows a dialog box titled "Field Configuration Wizard" with a close button in the top right corner. The main heading is "Database Column" with a sub-instruction "Specify the column name and size." and a small circular icon with a red dot. Below this, a paragraph explains that the column name will be automatically created. There are two input fields: "Column Name:" containing the text "DAY" and "Column Size:" containing the number "6400". A note at the bottom explains that column sizes are auto-calculated and not editable. At the bottom right, there are three buttons: "<Back", "Next>", and "Cancel".

Field Configuration Wizard

Database Column
Specify the column name and size.

Specify the column name that this Field maps to in the Dataset table. This column will be automatically created (as varchar or similar) in supported data providers.

Column Name:

Specify the column size.

Column Size:

Note: If you chose to Auto-Calculate column sizes in the Dataset configuration, you will not be able to edit the column size of individual columns unless you unselect that option on the Dataset Properties page.

Click 'Next' to continue.

<Back **Next**> Cancel

Specify the name of the column in the datastore. In the wizard, names will be defaulted to the Field name in all caps. If you chose to Automatically calculate the column sizes in the Dataset configuration wizard, the Column size will not be editable. Also, because the column sizes are calculated to consume the maximum row size of the specified datastore, the column size for each field is recalculated each time a new field is added; Therefore, the column size you see now is probably not the column size after all the fields are added. Click 'Next>' to continue.

Optimize Extraction Tags

Field Configuration Wizard

Extraction Tag Optimization

Adjusting the Field extraction tags is one of the most effective ways of optimizing the precision of the overall datapage extraction routine.

Extraction Tags are variable length strings of underlying source that immediately precede and follow the selected template field. The important thing right now is to make sure that neither tag contains data that may change from row to row (dynamic data). Later you may wish to tweak the tags to optimize the accuracy of the overall extraction template.

Start Tag

Min. Length:

Tag: -1>

End Tag

Min. Length:

Tag: </

-1>Wed </

<Back Next> Cancel

Web Scraper Plus+ automatically calculates extraction tags that will work to extract the template field from within the template dataset. It is possible, however, that you may need to optimize the tags to work on all data rows in all web pages. Until you test though, it is recommended that you use the default tags until you find a reason to change them. For now, just make sure that the tags don't contain any dynamic or content data. Click 'Next>' to continue. Click 'Finish' to complete the Field Configuration Wizard.

Add Additional Fields or Finish

Add Additional Data Fields
Click 'Add Another Field' to create an additional data field in the dataset. Click 'Next' to continue.

Keep adding additional data fields until you have accounted for all data in the template data item (row). Click 'Add Another Field' to create a new data field.

Current Items:

- Yahoo Finance News
 - Recent News
 - Fields
 - Day

To continue without adding an additional field, click Next.

[Add Another Field](#) [Next >](#) [Close](#)

New research reports for **CSCO**

Wed	2:52pm	CSCO	Riverstone to supply routers to Reuters
Wed	2:18pm	CSCO	[\$\$ - free trial] Cisco Falls. The News - RealCommentary from TheStreet.com
Wed	9:10am	CSCO	[external] A 99-Cent Hamburger
Wed	9:09am	CSCO	[external] MARKET MOVERS: Missing Estimates - at BusinessWeek.com
Wed	9:03am	CSCO	[\$\$ - free trial] Rally's on Its Last Legs - RealCommentary from TheStreet.com
Wed	8:07am	CSCO	Tokyo Stocks Drop to 19-Year Low
Tue	10:15pm	CSCO	Tokyo Stocks Fall 54 Points - Reuters
Tue	7:31pm	CSCO	Chief Execs Sour on Next Year
Tue	6:35pm	CSCO	UPDATE 1-After the Bell-Northern Networks Appliance up - Reuters
Tue	6:30pm	CSCO	[\$\$ - free trial] Faith in Tech Trailing

<TABLE cellPadding=2 border=0><!-- Yahoo TimeStamp: 103721712 vAlign=top><TD noWrap>Wed </TD></TABLE>

Click 'Add Another Field' to return to step "7.1 Select the Field" or click 'Next' once you have added all the fields.

Test the Datapage

Completing the Create Datapage Wizard

You have successfully created a Datapage extraction template! Leave the 'Run Test Extraction' checkbox checked to see the results of the datapage execution. You can manually fine tune the Datapage using the Datapage Editor after this wizard completes. Click Finish to complete.

Run Test Extraction

[< Back](#) [Finish](#) [Close](#)

New research reports for [CSCO](#)

Wed 2:52pm CSCO [Riverstone to supply routers to Reuters](#)

Wed 2:18pm CSCO [\[\\$\\$ - free trial\] Cisco Falls, The News - RealCommentary from](#)

Wed 9:10am CSCO [\[external\] A 99-Cent Hamburg TheStreet.com](#)

Wed 9:09am CSCO [\[external\] MARKET MOVERS: Missing Estimates - at Busines](#)

Wed 9:03am CSCO [\[\\$\\$ - free trial\] Rally's on Its La RealCommentary from TheStre](#)

Wed 8:07am CSCO [Tokyo Stocks Drop to 19-Year Press](#)

Tue 10:15pm CSCO [Tokyo Stocks Fall 54 Points - /](#)

Tue 7:31pm CSCO [Chief Execs Sour on Next Yea](#)

Tue 6:35pm CSCO [UPDATE 1-After the Bell-Nords Network Appliance up - Reuter](#)

Tue 6:30pm CSCO [\[\\$\\$ - free trial\] Faith in Tech Tr RealCommentary from TheStre](#)

Riverstone to supply routers to Korea's Hanaro </FONT

After you have added all the fields in the dataset and have otherwise finished the creation of your datapage, you can now test to see whether the extraction works. Simply leave the 'Run Test Extraction' checkbox checked and click 'Finish'. If you uncheck the 'Run Test Extraction' checkbox, the test extraction will not be run, but you can run it at a later time by selecting Datapage>Test Extract.

Extracted						
10 Items Extracted						
DAY	TIME	SYMBOL	ARTICLE_URL	HEADLINE	META_SRC_URI	META_PKG_ID
Wed	2:52pm	CSCO	http://biz.yahoo...	Riverstone to su...	Unavailable	N/A
Wed	2:18pm	CSCO	http://biz.yahoo...	[\$\$ - free trial] Ci...	Unavailable	N/A
Wed	9:10am	CSCO	http://rd.yahoo...	[external] A 99-C...	Unavailable	N/A
Wed	9:09am	CSCO	http://rd.yahoo...	[external] MARK...	Unavailable	N/A
Wed	9:03am	CSCO	http://biz.yahoo...	[\$\$ - free trial] R...	Unavailable	N/A
Wed	8:07am	CSCO	http://biz.yahoo...	Tokyo Stocks Dr...	Unavailable	N/A
Tue	10:15pm	CSCO	http://biz.yahoo...	Tokyo Stocks F...	Unavailable	N/A
Tue	7:31pm	CSCO	http://biz.yahoo...	Chief Execs Sou...	Unavailable	N/A
Tue	6:35pm	CSCO	http://biz.yahoo...	UPDATE 1-After...	Unavailable	N/A
Tue	6:30pm	CSCO	http://biz.yahoo...	[\$\$ - free trial] Fa...	Unavailable	N/A

Compare the results of the Test Extraction with the actual data on the web page. If the results match the data, then you have a perfect extraction template. If some data is missing or some extraneous data is included, you will need to optimize the extraction template probably by editing field tags or setting some fields as optional.

If you wish to print the extraction results, you may by clicking the Print button, otherwise click close.

Optimizing Heuristically Adaptive Extraction

After testing a datapage extraction template, you may find that you can improve the accuracy and precision of the extraction by modifying various attributes of the template. There are three such attributes at your disposal to tweak extraction performance: Dataset Initialization String, Field Start and End Tags, and the Optional Field Flag.

Comment [Help1]: Edit Field Start and End Tags

Comment [Help2]: Optional Field Flag

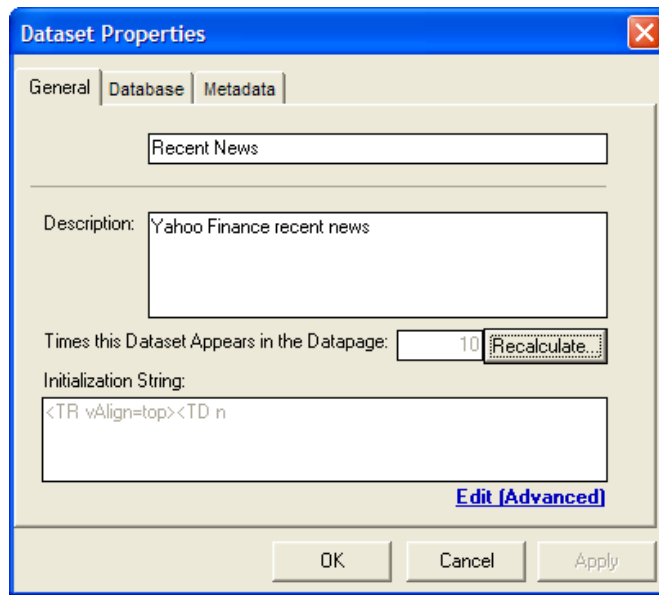
Before altering a datapage, it will help if you have a thorough understanding what happens when a datapage is run. You may wish to create a copy of the existing datapage so that you have a baseline to refer to.

Comment [Help3]: What Happens When a Datapage Executes

Comment [Help4]: 1.Launch the Datapage Editor from the Providus Console

Edit Dataset Initialization String

You can modify the Dataset initialization string in two ways. You can re-select the number of datarows in the template Dataset and recalculate, or you can manually specify the Initialization string. Both of these options are accessible through the Dataset Properties dialog.



To recalculate the Initialization String, click the “Recalculate...” button. Web Scraper Plus+ will automatically calculate an exhaustive list of the possible number of datarows on the template page and display a dialog containing this list. Select the appropriate number of occurrences and click OK.

You can also manually edit the Initialization String by clicking the “Edit (Advanced)”. You may wish to manually enter an Initialization string when you are certain that you know what the Initialization string is. For example, if you wish to get all of the hyperlinks on a page, you may want to manually enter ‘href=’. Note that initialization strings (as well as start and end tag for fields) are case insensitive, so HREF is the same as Href is the same as href.

Edit Field Start and End Tags

Perhaps the most frequently used means of tweaking the precision of a datapage is editing Field Start and End Tags. When a datarow is extracted each field is extracted in the order of its appearance in the template dataset and optionally not extracted based on whether it is set as optional. The value of the field comes from the underlying source found between the end of the start tag and the beginning of the end tag.

Web Scraper Plus+ will not allow you to specify a start tag that includes data from either before the beginning of the dataset initialization string or before the beginning of the previous field’s end tag. Web Scraper Plus+ will also not allow you to specify an end tag that includes data beyond the end of the next field’s start tag. Any tag that meets these restrictions is allowed, however. Web Scraper Plus+ automatically selects the shortest possible start tag that both meets the above requirements and is unique between the last previous field and the current field. It automatically selects

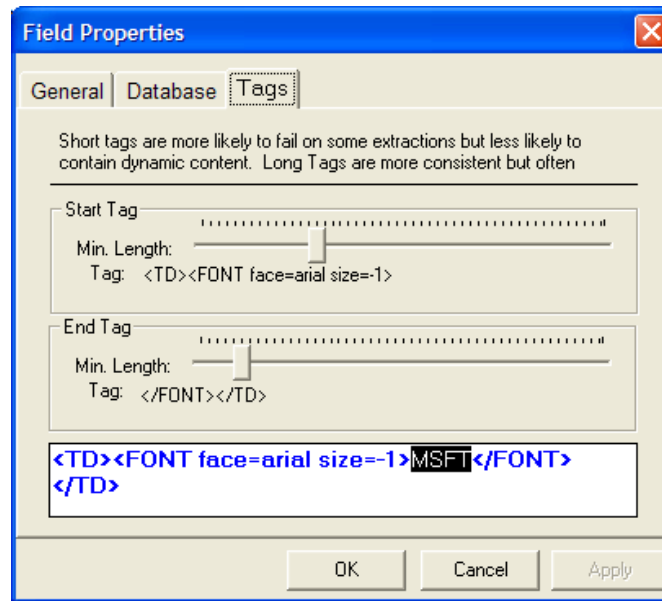
the shortest possible end tag such that the above requirements are met and the end tag does not occur within the template field's value.

You can increase or decrease the length of a field's start or end tags keeping in mind the following generalizations:

Increasing tag length makes it less likely that your extraction will mistakenly identify the beginning or end of your field, thus returning invalid data.

Decreasing tag length makes it less likely that your tags will inadvertently include dynamic data not contained in all datarows, thus returning invalid data.

So, while either increasing or decreasing tag length can benefit the accuracy and precision of your extraction, both also carry potential risk. For best results, make the start and end tags as long as possible without including any dynamic data.



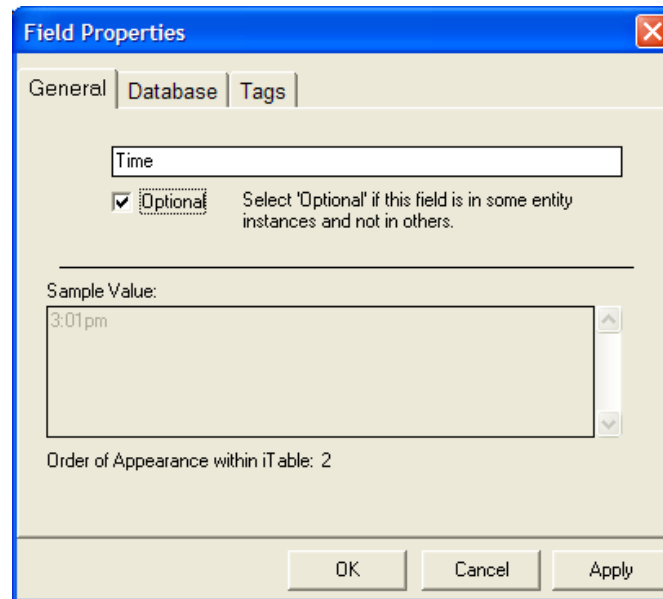
Slide the tag length indicator to the right to make the tag longer, to the left to make it shorter. The text area below shows the Start Tag in blue followed by the field value highlighted and finally, the end tag in blue.

Optional Field Flag

Sometimes datasets are not completely predictable. For example, every item on eBay does not have a "Buy It Now" price, but some do. Traditional extraction techniques fail under these circumstances. Web Scraper Plus+ does not. It uses patent pending Heuristically Adaptive Extraction Algorithms (HAXA) that allow you to set data fields as optional. This way if the dataset is not completely homogeneous, the extraction will still return extremely good results

When a datapage is executed on a specific web page it looks at the underlying source of the page for the first instance of the Initialization String for one of its Datasets. It

then creates an internal structure of all the combinations of fields (for the Dataset of the first Initialization String found) based on whether a field is optional. Then, from the location of the Initialization String forward, each combination of fields is tested and a proprietary scoring algorithm is applied to determine most precise extraction. Using the combination of optional and required fields with the most precise extraction, each field in the dataset is extracted by moving forward in the underlying source from the dataset Initialization string.



You may find after testing an extraction that a field you originally thought was required is, in fact, optional. Alternatively, you may find that a field originally thought to be optional is required. Through the field properties dialog (General Tab), you can change a field's optional flag to reflect this change.

Why not just make all fields Optional? Since Web Scraper Plus+ needs to test each combination of optional and required fields prior to actual extraction to determine the most precise combination, fewer optional fields dramatically improve performance. In fact, making a field required instead of optional can nearly double the performance of the actual extraction logic. Depending on the size of your extraction, this may or may not be significant as extractions are typically measured in pages/sec, but anytime you can potentially double your performance, it is useful to at least note.

What Happens When a Datapage Executes?

A datapage is the root of an extraction template. As its name suggests, it maps directly to a webpage. A datapage consists of one or more datasets. Datapages can be linked together to "walk" entire site hierarchies through the use of packages.

When a datapage is executed on a specific web page it looks at the underlying source of the page for the first instance of the Initialization String for one of its Datasets. It then creates an internal structure of all the combinations of fields (for the Dataset of the first Initialization String found) based on whether a field is optional. Then, from the location of the Initialization String forward, each combination of fields is tested and a proprietary scoring algorithm is applied to determine most precise extraction. Using the combination of optional and required fields with the most precise extraction, each field in the dataset is extracted by moving forward in the underlying source from the dataset Initialization string. The field value comes from what (in the underlying source) lies between the field's start and end tag. Each field in the dataset is extracted and loaded into the target datasource.

Success-Failure data is calculated based on an aggregation of the scoring algorithms and the ratio of the number of times a dataset's initialization string was found and the number of datarows actually extracted for that dataset.

Link Datapages Together Using Download Tasks

If you wish to link Datapages together such that you can walk the hierarchy of a website, extracting all the pertinent data along the way, you can do with a Web Crawler, or in some cases more precisely by extracting a url from a datapage then using that column to supply the file list for another task

For example, EBay Motors has a hierarchy of listings that can be walked in the following order: Type→Make→Model→Item Summary→Item Detail. To walk this hierarchy, a datapage needs to be created for each of these page types. The URL linking each level to the next highest level must be extracted along with any additional information. So, the Make datapage must extract the Model_URL; the Model datapage must extract the ItemSummary_URL; and so on. Then, a task is created to extract each datapage. Each task is configured to return a list of URL's from the table that contains its parent. So, the Model Task will be a SQL Task that points to the Make table and returns all the URLs that point to pages that can be extracted by the Model datapage (SELECT MAKE_URL FROM EBAY_MAKES). It follows that the ItemSummary Task contains the ItemSummary datapage and its SQL would look like SELECT MODEL_URL FROM EBAY_MODELS.

This process of linking can continue forever, the only obvious point is that the tasks must be ordered properly in the package such that parent tasks are extracted before children. It is also useful to note that you are not limited to SELECT statements to create your lists of Urls but anything you can do from SQL, so stuff like

```
SELECT 'http://listings.ebaymotors.com' +  
REPLACE(ITEM_URL, '&', '&') AS ITEM_URL  
FROM EBAY_LISTING
```

also works.

Once You have Created a Package...

Import/Export

Import/Export a Datapage

Datapages can be exported to file format and imported back into Web Scraper Plus+ very easily. This allows for the simple and straight forward development, distribution, sharing, and archival of Datapage templates. To export a datapage, simply select Datapage>Export from the console. The default file extension is .dtp.

To import a Datapage, select Datapage>Import from the console and select the datapage from the Open file dialog.

Remember that your datapage may contain sensitive account information. Although this information is encrypted using AES 256 bit encryption, it is recommended that you nullify the account information if you are distributing the datapage in an untrusted environment.

Import/Export a Package

Like datapages, Packages can be exported to file format and imported back into Web Scraper Plus+ very easily. Packages contain all their constituent Tasks and dependent datapages. This allows for the simple and straight forward development, distribution, sharing, and archival of Packages. To export a Package, simply select Package>Export from the console. The default file extension is .ppk.

To import a Package, select Package>Import from the console and select the Package from the Open file dialog.

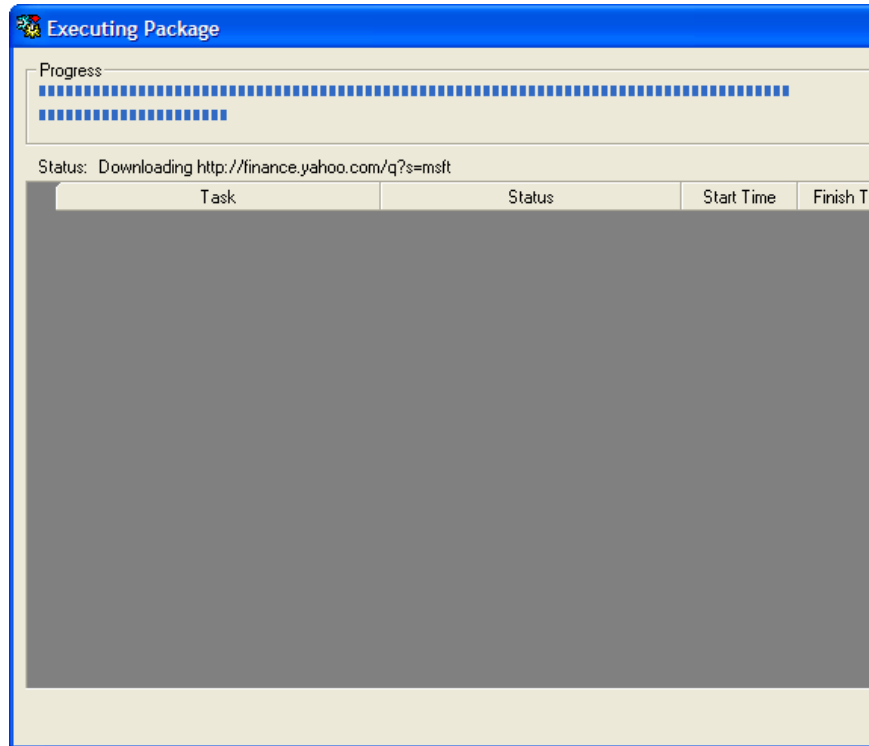
Remember that your datapages and tasks may contain sensitive account information. Although this information is encrypted using AES 256 bit encryption, it is recommended that you nullify the account information if you are distributing the datapage in an untrusted environment.

Package Execution

Packages can be executed either from the console or via the command line. The benefit of command line execution is that it can be run scheduled using Windows Scheduler or an Enterprise management tool such as CA Unicenter.

Execute a Package from the Console

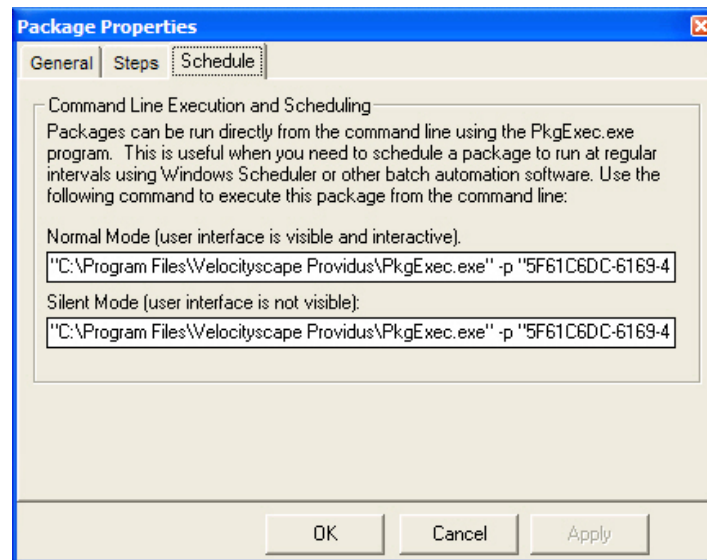
To execute a package from the console simply select a package, then select Package>Execute from the menu. The package execution window will load, and the package will execute.



Scheduling a Package to Run Unattended

Web Automation Packages can be scheduled to run using Windows Scheduler or any commercial batch scheduling suite including products by Tidal, BMC, and Computer Associates. One pretty decent batch scheduling product for small and mid-sized businesses is ActiveBatch (<http://www.activebatch.com>) Every Web Scraper Plus+ Package can be scheduled to run via the command line. Using the command line, the packages can be configured to run with or without a user interface. The command line works exactly the same as clicking execute from the console. All you have to do to schedule a package to run at a certain time (or due to a specific event like the

arrival of an email or a change to a file or directory) is enter the command line into the batch definition in whichever batch scheduler you choose.



Execute a Package from the Command Line

Packages can be executed from the command line using the PkgExec.exe executable. This is useful for scheduling commonly run tasks. In this way, Web Scraper Plus+ can integrate with Windows Scheduler or a variety of enterprise management applications such as CA Unicenter, IBM Tivoli, and BMC.

The syntax for PkgExec.exe is:

```
PkgExec -p PackageID [-s]
```

Parameters

- p PackageID Specifies the Package to be executed by its ID.
- s Executes the Package in Silent (No UI) mode.
- ? Shows this information.

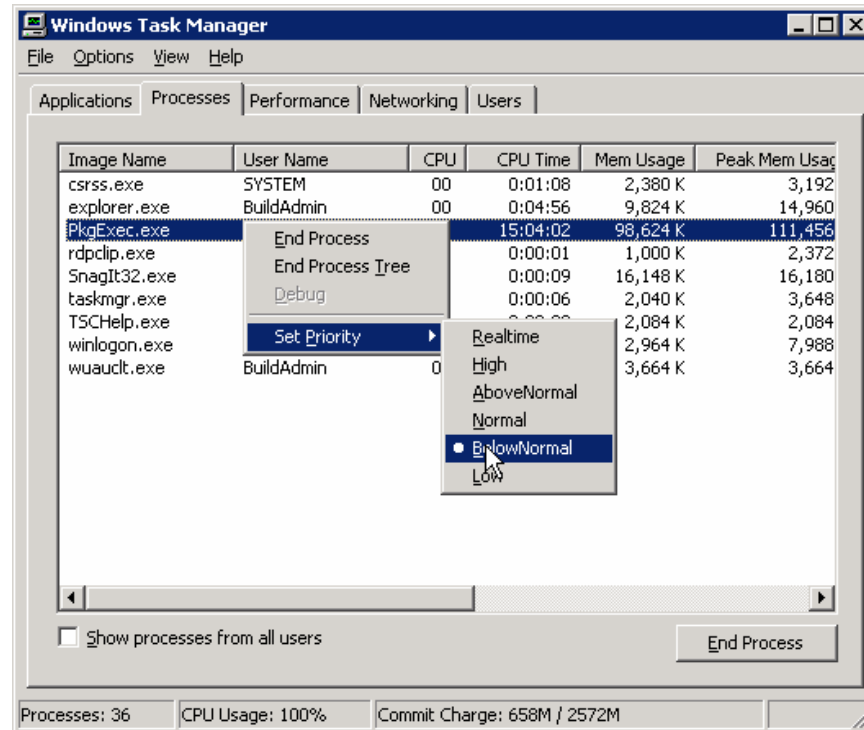
Example

```
PkgExec -p "C67AEF84-4CD4-A17A-B7EC35A52924" -s
```

You can find the ID of the Package on the general tab of the Package Properties window.

Running a Package in the Background

Web Automation Packages often take hours or days to complete, and you may not always have the luxury of running it on a separate server dedicated solely to the task. In such a case, you can set the package to run in the background so that it will only consume the CPU when you are not using it.

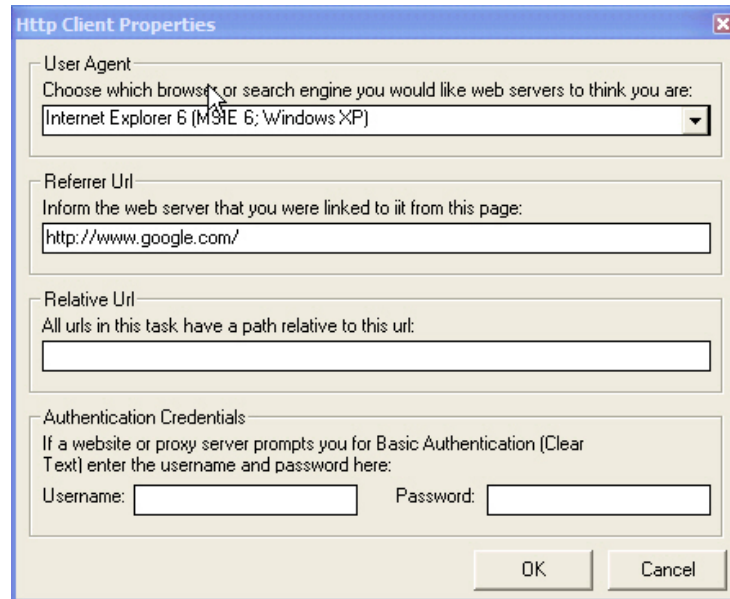


To run an executing package in the background, open Task Manager (ctrl+alt+del). Right-click the running PkgExec.exe process on the Processes tab. Select 'Set Priority' and choose BelowNormal, or even Low from the list. Click past the warning (don't worry, the package won't crash). Voila! The package is will now consume your CPU only when another application does not need it. The package will run just as fast overnight (or whenever it is idle) but, will not max out your CPU while you are trying to work.

Http Client Configuration

All of the downloads, form posts, and crawling that you can do with Web Scraper Plus+ is performed by an army of Http clients called user agents. To a web server these user agents look just like a normal browser (you can even make the web server believe you are Internet Explorer, Netscape, or even the Google web crawler, but more on that later.) In fact, the agents are not web browsers at all. If you tried to download 5+ pages per second over any length of time with your normal web browser, it would crash pretty quickly. Browsers just aren't built for the rigors of industrial grade automation. Web Scraper Plus+ is built on a server grade Http stack called WinHttp. For more information on WinHttp visit: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/winhttp/http/winhttp_start_page.asp .

The Web Scraper Plus+ user agents handle sessions, cookies, and redirects just like a normal web browser. Furthermore, the sessions and cookies obtained from one task are passed along to the next task in a package, so once you login to a site in one task you remain logged in until the package completes, or you log off. However, because the agents are not based on your web browser, it does not share the persistent cookies that you may obtain while casually browsing the web. So, every time you run a package it is as if the agents are accessing the site for the first time. From a scalability perspective, that is really a good thing, because it means, for example, that you can log in to a web site with multiple different users without worrying that the page that the agent is seeing is cached or it is confused as to your identity. In essence what it lacks in some convenience it makes up for in the scalability and predictability of the system.



Static Referrer Url

Sometimes a web server will check to see what url linked your browser to the page that you are posting a form to or downloading a page from. In some cases, if you do not provide the correct referrer url, then you will receive an error. Webmasters do this for two primary reasons: security and marketing. Web Scraper Plus+ allows you to set any referrer url that you would like for each task. By default, when you set up a form task, the Referrer Url will be set to the page that the form is on. That should be the correct setting for pretty much all forms. Remember that all agent properties are inherited by future tasks, so if your first task is a login form, you may receive mixed results if your next task uses the same referrer url.

Impersonating Web Browsers and Search Engine Crawlers

Web servers can identify the type of browser that is accessing their site and webmasters can look into their logs and see when Google crawled their site last. The reason that this is possible is because user agents (browsers and crawlers) identify themselves when making a request. For example, Internet Explorer running on Windows XP would identify itself as "Internet Explorer 6 (MSIE 6; Windows XP)", whereas the Google's search crawler identifies itself as "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)". You can impersonate either of these or you can create your own moniker.

In most cases it is probably best to stick with the default of IE6, because sometimes websites change their content depending on the browser, and if you create a datapage it is based on IE6 content. Also, some sites don't want their content crawled by search engines, so they may turn you away if you identify yourself. In other cases, although this has never happened to us or any of our clients (to our knowledge), a webmaster looking at her log may see 10,000 requests coming from the same IP address and decide to block your IP address. Most likely she will assume that your IP is a proxy server for a large organization. But, if you impersonate the googlebot she will quite certainly assume you are Google, and will not block you (unless of course she does not want her site googled).

Browser Authentication

There are two types of browser authentication supported by most browsers: Basic Authentication (Clear Text) and Windows (NTLM) Authentication. Basic Authentication is supported by most internet web servers, but Windows Authentication is primarily supported by Microsoft IIS and is cannot be used to authenticate over the internet. NTLM authentication is used fairly frequently on Intranets. Web Scraper Plus+ supports Basic Authentication and it can be configured in the agent properties window. Web Scraper Plus+ does not support NTLM authentication.

If you find that you need to access a site that requires Windows Authentication or if you are running Web Scraper Plus+ behind a Microsoft Proxy or IAS server that requires NTLM authentication then you still have options. You can set up a local proxy on the Web Scraper Plus+ machine that translates normal requests and sends them through the MS Proxy with NTLM credentials. There is a free one located at <http://ntlmmaps.sourceforge.net/>. It is aptly named "NTLM Authentication Proxy Server".

Bandwidth Throttling

Web Scraper Plus+ can issue 4-10 requests per second. This amount of load can bring a small web server to its knees. You may wish to throttle the amount of bandwidth you use. You can do this on a task by task basis by editing a task once it is created.

