

Web Crawler

Scraper Manual

David Tee



06

Contents

Scraper Overview	3
Tool Bar	3
View Node Data Window	4
Result Data Window	4
Result Data Window w/ Multiple Tables	5
Tag Tree.....	5
Tag Properties	6
Sample [Scrape] Projects	7
Basic Tag.....	7
Basic Tag 2.....	7
Regex Tag	7
Regex Data	8
Multi Tag	8
Look Behind.....	8
Optional Tag.....	9
Max Data Length	10
Sample Project: BackCountry.com.....	11
Sample Crawl Project: Lyrics007.com	Error! Bookmark not defined.

Scraper Overview

The scraper allows the user to configure a setting to extract/parse the data from strings that contains similar formatting.

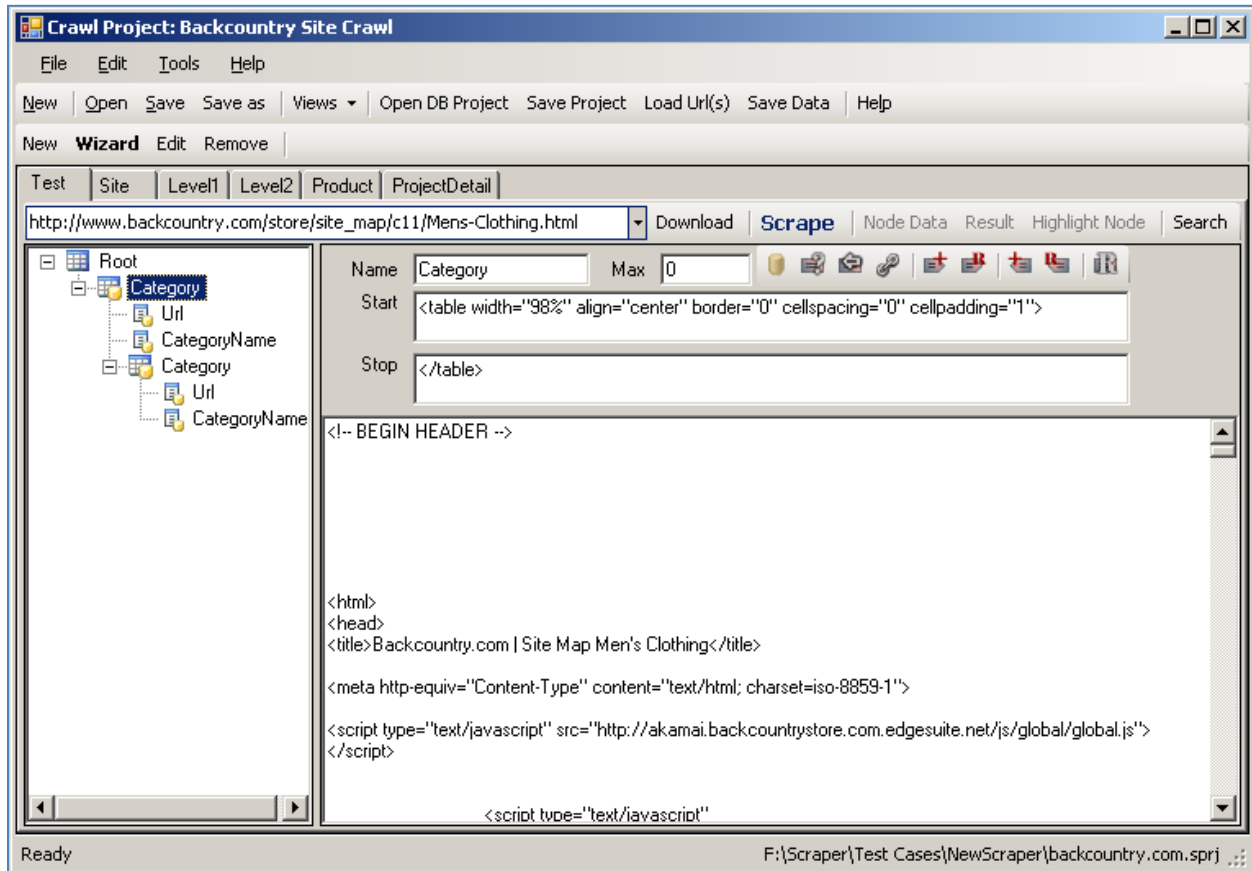


Figure 1: Scraper

Tool Bar

Tool bar contains buttons to download context from the web address given. It also allows scraping of the page based on the tag tree.

Toolbar Item	Description
Url List Combo box	User can enter in url here to download a url.
Download	Download the context of the url to the sample text box .
Scrape	Scrape the current context in the sample text box.
Node Data	View the scraped data of the selected tag tree. This will toggle the "View Node Data" Window. This button is only available after the Scrape is successful.
Result	View the resulting dataset of the scrape. This is the dataset scraped from the current tag tree. This button is only available after the Scrape is successful, and the scrape generate at least one data table.
Highlight Node	Highlight the text scraped.

Search	Search the pattern/rules defined in the tag properties
---------------	--

View Node Data Window

This window allows the user to view the resulting data from the current node selected in the Tag Tree

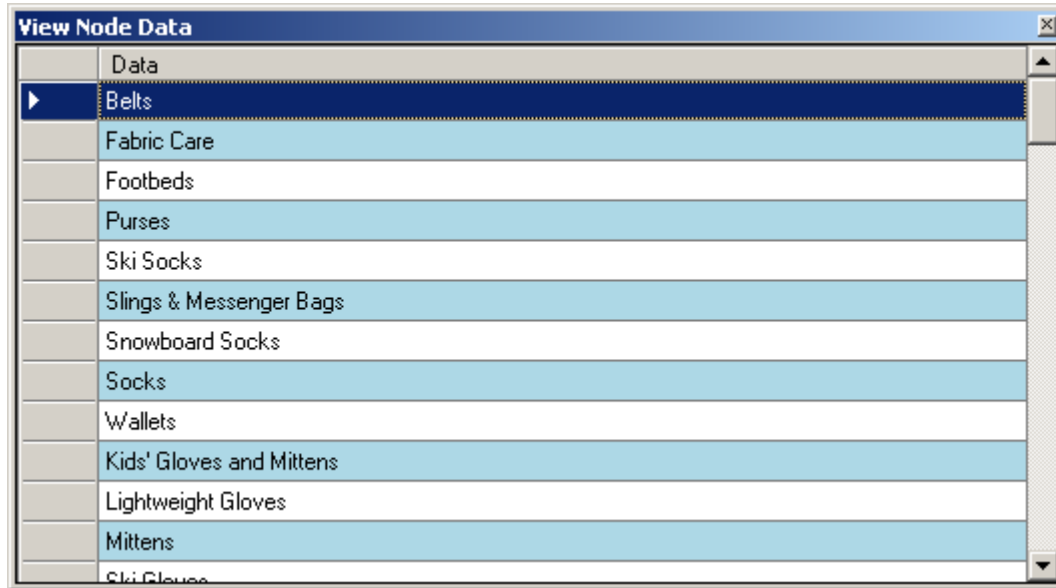


Figure 2: View Node Data Window

Result Data Window

This window allows the user to view the resulting dataset from the scrape. (The resulting dataset is only available if the tag library generates at least one data table.) In this window, you can select the table to view (if multiple tables are generated) , and the child table if there's any.

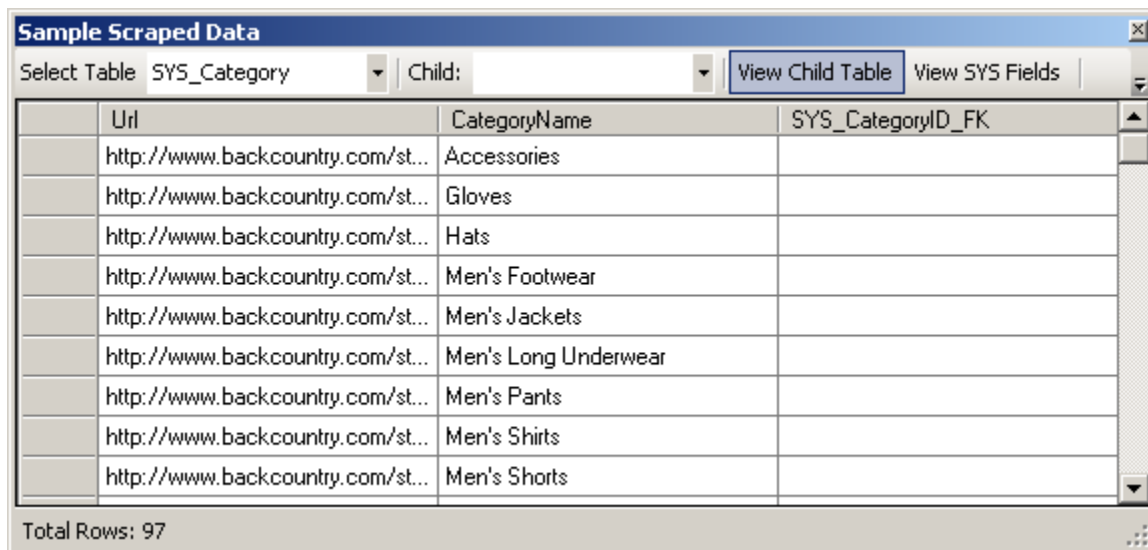
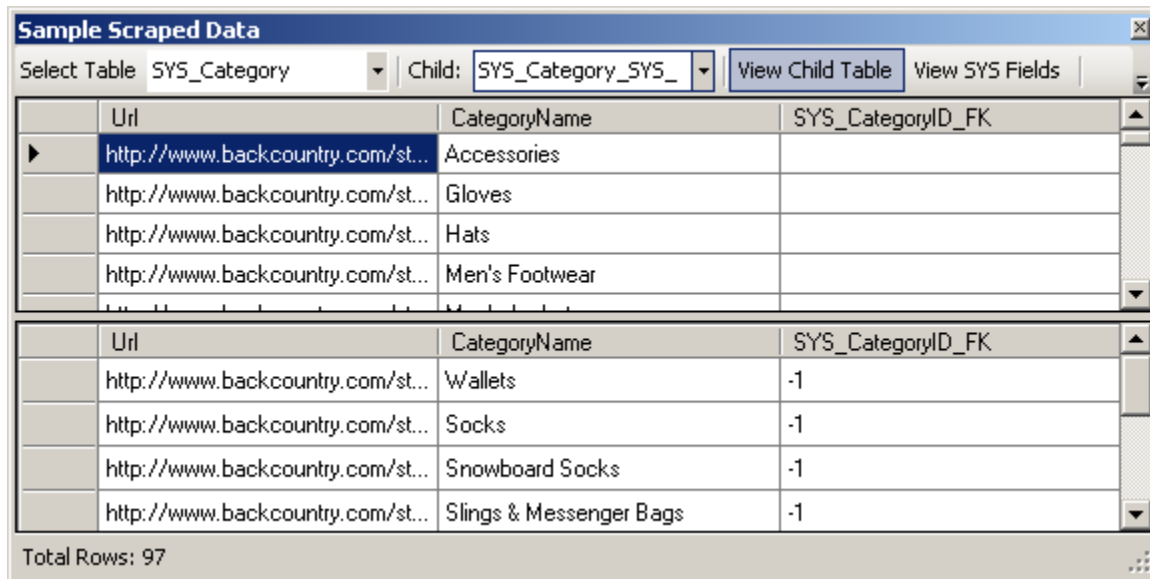


Figure 3: Result Data

Result Data Window w/ Multiple Tables



Sample Scraped Data

Select Table: SYS_Category | Child: SYS_Category_SYS_ | View Child Table | View SYS Fields

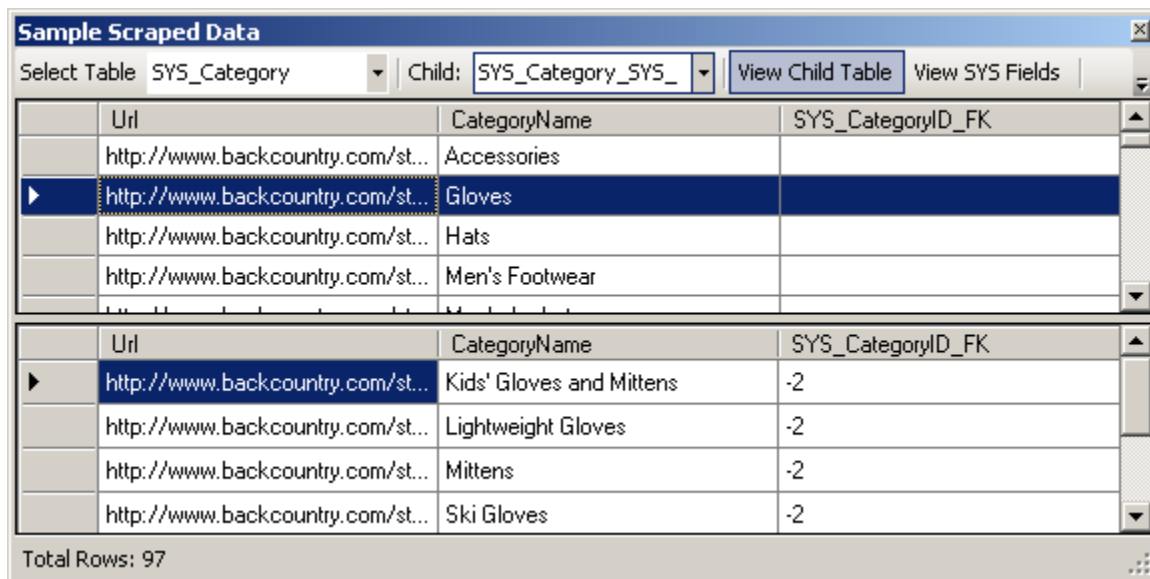
Url	CategoryName	SYS_CategoryID_FK
http://www.backcountry.com/st...	Accessories	
http://www.backcountry.com/st...	Gloves	
http://www.backcountry.com/st...	Hats	
http://www.backcountry.com/st...	Men's Footwear	

Url	CategoryName	SYS_CategoryID_FK
http://www.backcountry.com/st...	Wallets	-1
http://www.backcountry.com/st...	Socks	-1
http://www.backcountry.com/st...	Snowboard Socks	-1
http://www.backcountry.com/st...	Slings & Messenger Bags	-1

Total Rows: 97

Figure 4: Result Data w/ Multiple Tables

In the figure above, you will notice that the view contains two tables. The table on top contains the first table, and the table on the bottom is the child table (a table that have foreign key constraint that that references another table). Changing the row selected in the top table will change the content of the child table.



Sample Scraped Data

Select Table: SYS_Category | Child: SYS_Category_SYS_ | View Child Table | View SYS Fields

Url	CategoryName	SYS_CategoryID_FK
http://www.backcountry.com/st...	Accessories	
http://www.backcountry.com/st...	Gloves	
http://www.backcountry.com/st...	Hats	
http://www.backcountry.com/st...	Men's Footwear	

Url	CategoryName	SYS_CategoryID_FK
http://www.backcountry.com/st...	Kids' Gloves and Mittens	-2
http://www.backcountry.com/st...	Lightweight Gloves	-2
http://www.backcountry.com/st...	Mittens	-2
http://www.backcountry.com/st...	Ski Gloves	-2






Total Rows: 97

Figure 5: Result Data view w/ different row selected

Tag Tree










Tag tree allows the tree view of the tags in the tag library. Tag Tree represents the tree of tags. This could be compared to the DOM tree. User can move drag and drop the tag node into another node like

can in window's file explorer. User can also move the the tag node up or down to position the scrape patterns.

Icon	Item	Description
	Table and Field	The tag field will be saved to database, but it also serves as a temporary field for extraction data.
	Temporary Table	Tables that will be saved to database, this is probably an intermediate tag node to narrow down the desired data.
	Temporary Field	Field that will not be saved to database, this is to filter out some of the fields that must be configured in order for the scraper to scrape other fields that would be saved to database.
	Table	Table that will be saved to database.
	Field	The field being saved to database.

Tag Properties

The tag contains description of the text that's the desired data to be extracted is bounded by. The example included with the program will further explain the meaning of the properties listed.

Icon	Property	Description
	Name Text box	The name of the tag, this must be unique with in a tag group.
	Max Text box	The maximum length of the characters of the data to be extracted.
	Save Data	Save the data as data field in the database
	Optional	Optional Tag
	Reverse Search	The search pattern will be searched in reverse
	Url Data	The data scraped is url link. The crawler will fix the relative url and make the absolute url based on the current url that's being scraped.
	Append Start Tag	Append the start tag to the data
	Regular Expression Start Tag	The start tag pattern is a regular expression. , the scraper will treat it as such.
	Append End Tag	Append the end tag to the data.
	Regular Expression End Tag	The end tag pattern is a regular expression, the scraper will treat it as such.
	Single Regular Expression	When checked, user may enter one regular expression pattern to search (There's no Start and End Tag text box)
	Start [Tag pattern] Text box	Start tag is similar to xml's elements. <xml>data</xml>. To extract the data, the start tag in this case on the <xml>
	End [Tag pattern] Text box	The end text that the desired data to be extracted is bound to. This can be described using regular expression or just literal text.

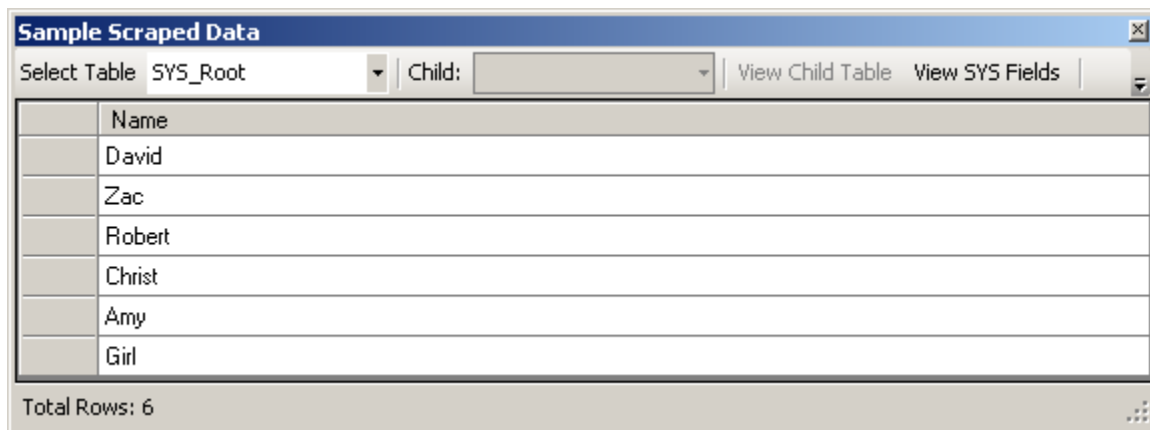
Sample [Scrape] Projects

The crawler is shipped with a sample crawl solution that contains sample Scraper Configurations. Open “example.sproj” for the sample data. The sample projects in with crawl solution is described in details below.

Basic Tag

This Scrape Project demonstrates the basic scraper configuration possible. On the tag library tree, there’s a root Node, named “Root”. It has one child node named “Name”. Because the child is a leaf node, the icon showed is a field icon. In database sense, Root will be the table and name would be one of the field of the table.

If the Save Data was enabled, then the field would be saved into database, and the icon for the Node will change, and a database table would be generated. Enable the save property of the Name tag and run the scrape again, you will see that the Result button is enabled. You can view the resulting data table of the current scrape project (the window below).



Name
David
Zac
Robert
Christ
Amy
Girl

Total Rows: 6

Figure 6: Result Data

In this project, the goal is to scrape the name from the given text. Name is bounded between <td> and </td> tags. So, in the Start text box and stop text of the name, I specified <td> as the start pattern, and </td> as the end pattern. The scraper will look for the following regular expression pattern: <td>(any characters)</td>.

Basic Tag 2

This is the same process as Basic Tag, but this time, there’s no start tag, but the end tag is one space. This demonstrates that a tag require either a start tag or end tag, nothing is specified in the start or end tag text box, then the scraper would return the text.

Regex Tag

In this Scrape project, the start tag is selected as a regular expression pattern, and the pattern specified in the start [tag pattern] text box is an regular expression that will match: < follow by any one digit, followed by td>.

Regex Data

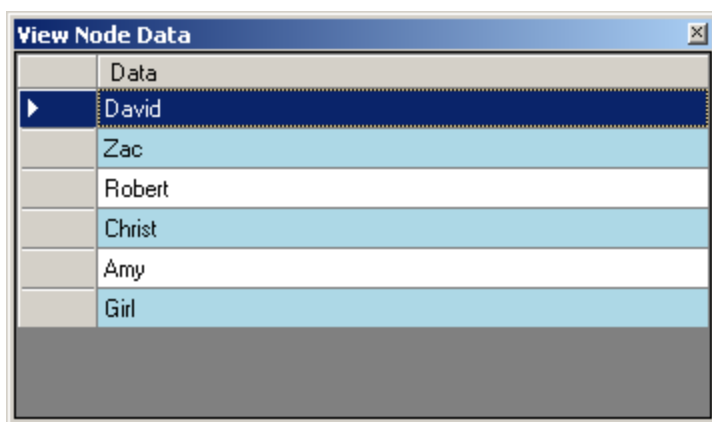
In this scrape project, only one regular expression pattern is defined. The scraper will search for the pattern (which looks for valid email address) and return the results.

Multi Tag

This scrape project demonstrates the multiple tag support in the tag library. The scraper will search for the pattern specified by multiple tags.

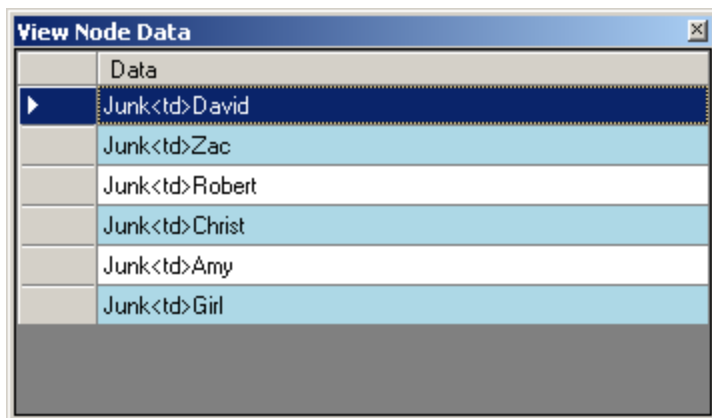
Look Behind

This scrape project demonstrates the look behind property of the a tag. Normally, the scraper will match the pattern going from left to right, but if look behind property of the tag is selected, the pattern is matched going from right to left.



	Data
▶	David
	Zac
	Robert
	Christ
	Amy
	Girl

Figure 7: Look behind enabled



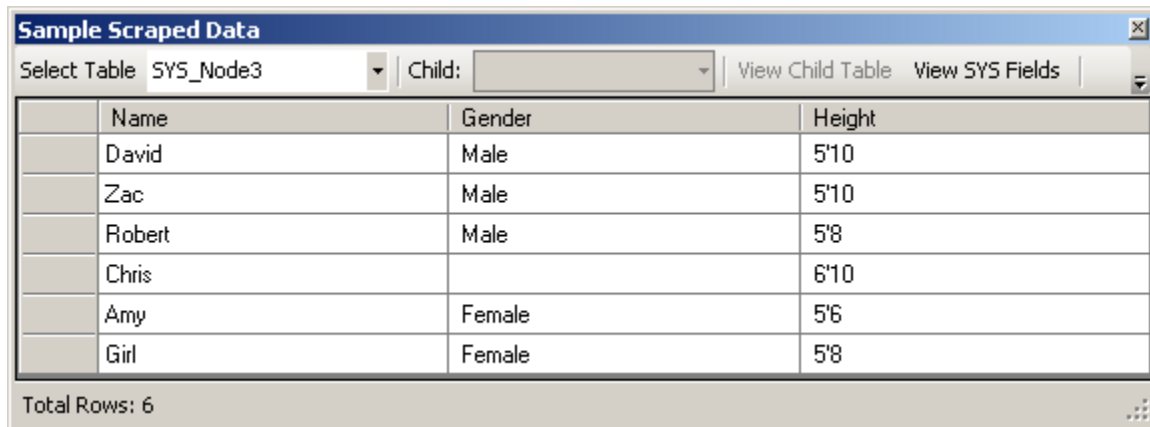
	Data
▶	Junk<td>David
	Junk<td>Zac
	Junk<td>Robert
	Junk<td>Christ
	Junk<td>Amy
	Junk<td>Girl

Figure 8: Look behind Disabled

When the look behind is disabled, the scraper will look for the following pattern: <td>(any characters)<\td>. If the look behind is enabled, the scraper will look for the same pattern, but run another pattern match on the result that would match the result going from right to left.

Optional Tag

This scrape project demonstrates the optional tag property. In the given sample data, you will find the pattern to be inconsistent. Named is bounded between <td> and
, Gender is bounded between
 and
, and the height is bounded between <td> and
. But in case of Chris, he is missing the Gender field, and we want the scraper to get Chris's information. The Optional Tag property of allows the scraper to scrape Chris even if one of the tags is missing. Try taking away the optional property, the result data would be different.



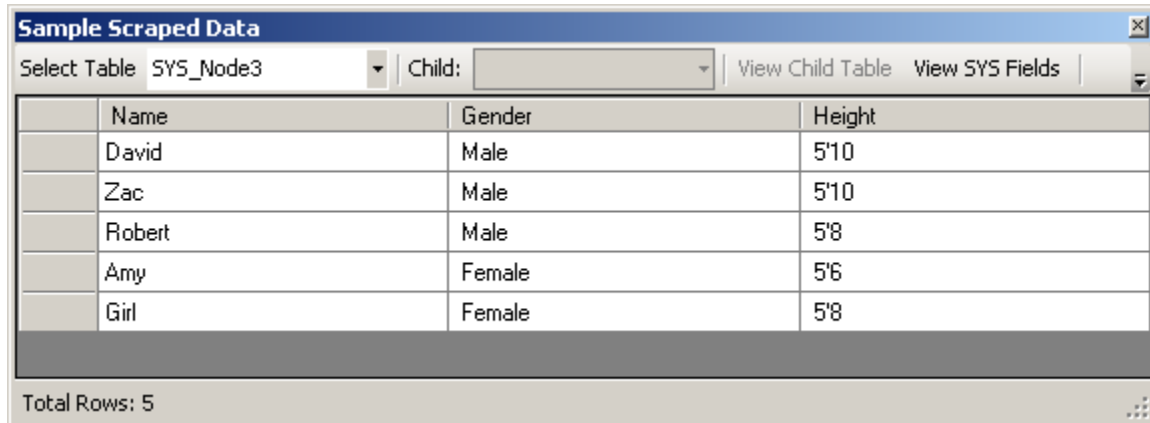
Sample Scraped Data

Select Table: SYS_Node3 Child: View Child Table View SYS Fields

Name	Gender	Height
David	Male	5'10
Zac	Male	5'10
Robert	Male	5'8
Chris		6'10
Amy	Female	5'6
Girl	Female	5'8

Total Rows: 6

Figure 9: With Optional Tag Enabled



Sample Scraped Data

Select Table: SYS_Node3 Child: View Child Table View SYS Fields

Name	Gender	Height
David	Male	5'10
Zac	Male	5'10
Robert	Male	5'8
Amy	Female	5'6
Girl	Female	5'8

Total Rows: 5

Figure 10: With Optional Tag Disabled

You will also notice that I placed a tag node (padded node) between the root and the fields that I want to scrape. If you look at the data scraped for the node, you will see the following:

	Data
▶	<td>David Male 5'10 </td>
	<td>Zac Male 5'10 </td>
	<td>Robert Male 5'8 </td>
	<td>Christ Male 6'10 </td>
	<td>Amy Female 5'6 </td>
	<td>Girl Female 5'8 </td>

Figure 11: Content of padded Node

If I take away this node, the result showed in the figure below, and it is not the data that I want the scraper to return. This is because the scraper searches for the pattern given and it found it.

Select Table	SYS_Root	Child:	SYS_Category_SYS_	View Child Table	View SYS Fields
	Name	Gender	Height		
	David	Male	5'10		
	Zac	Male	5'10		
	Robert	Male	5'8		
	Christ	6'10 </td> <td>Amy	Female 5'6		
	Girl	Female	5'8		

Total Rows: 5

Figure 12: Result without the padded Tag Node

Max Data Length

The max length properties allow the data scraped from the tag be bounded by character length. In this project, max size allowed is 4, so the scraper only scraped names that are less than 4 character length. David, Robert and Chris is ignored.

	Data
▶	Zac
	Amy
	Girl

Url

In this project, the concept of Url node is explored. A tag node that has url property enabled is a url node. The scraper will convert the relative url to absolute url after the scrape process. (This node allows the crawler to figure out the next url to download and scrape)

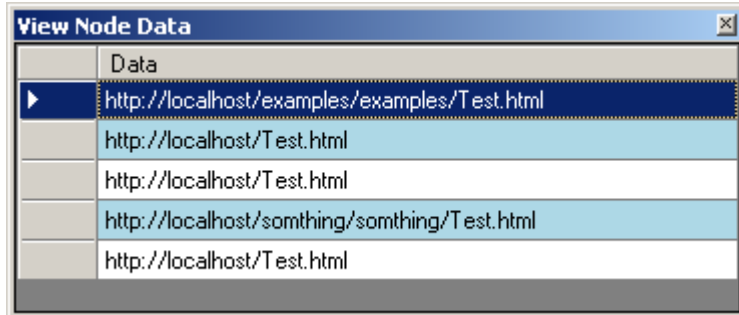


Figure 13: Result with Url Enabled

Complex 1

In this project, the concept of the scraper generating multiple tables with relationships is explored. Complex generates

Complex 2

This project is similar to Complex 1, but it generates three data tables.