

Scraping Web Pages for Data with the Web Viewer

FMUG

August 3, 2007

When to Scrape

- Just display the Web Viewer when all you need to do is view the information:
 - e.g., Contact information
 - Data does not become obsolete
- Scrape data from the Web Viewer when you need to store or manipulate data
 - e.g., report on scraped data, find an address on a map

What you need

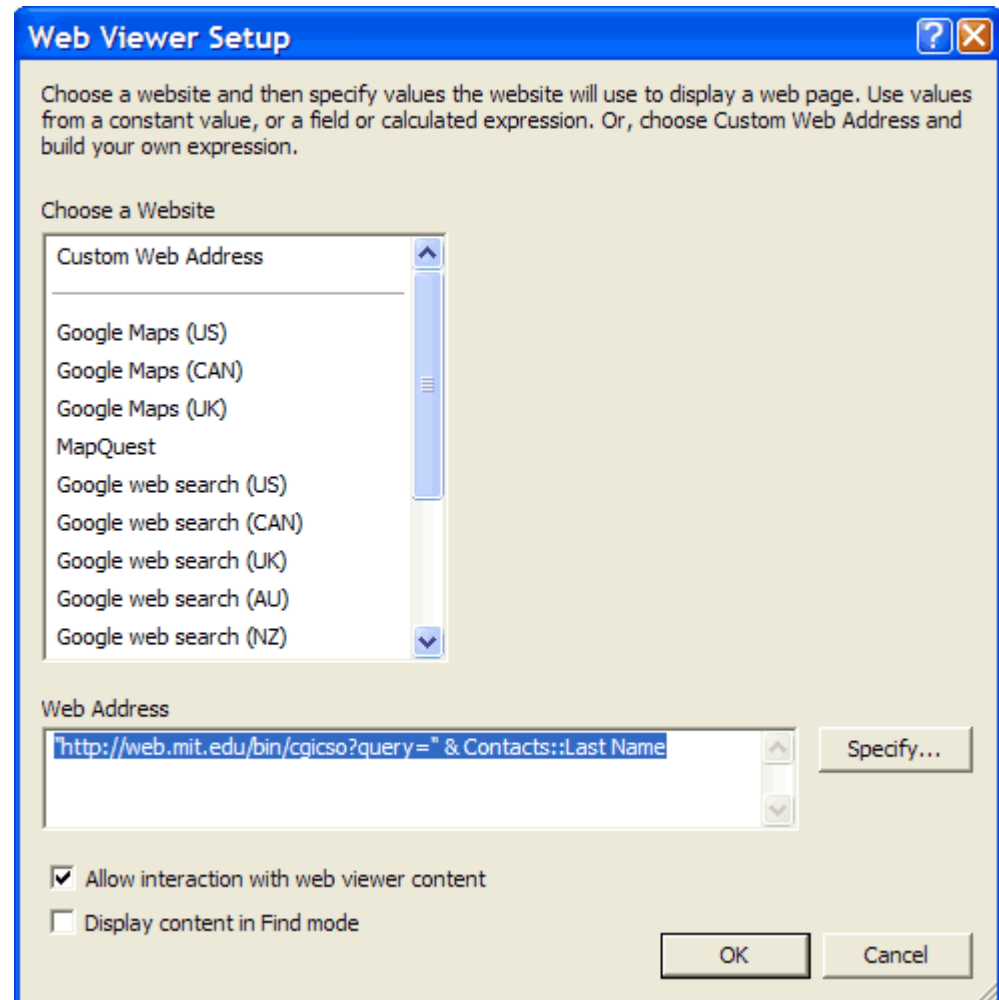
- FileMaker Pro v8.5
- Comfort with scanning web page source code
- Familiarity with using functions and variables
- Familiarity with creating scripts and calculations

Creating the Web Viewer Object

- Add a Web Viewer Object to your layout
- Give the object a name
- Create navigation buttons
- For details and screen shots, see <http://web.mit.edu/ist/help/filemaker/fmug/WebViewer.pdf>

Create Your Custom URL

- Double-click your Web Viewer object
- Enter your URL into the Web Address field
 - Paste from your browser
 - Build it by clicking the Specify button

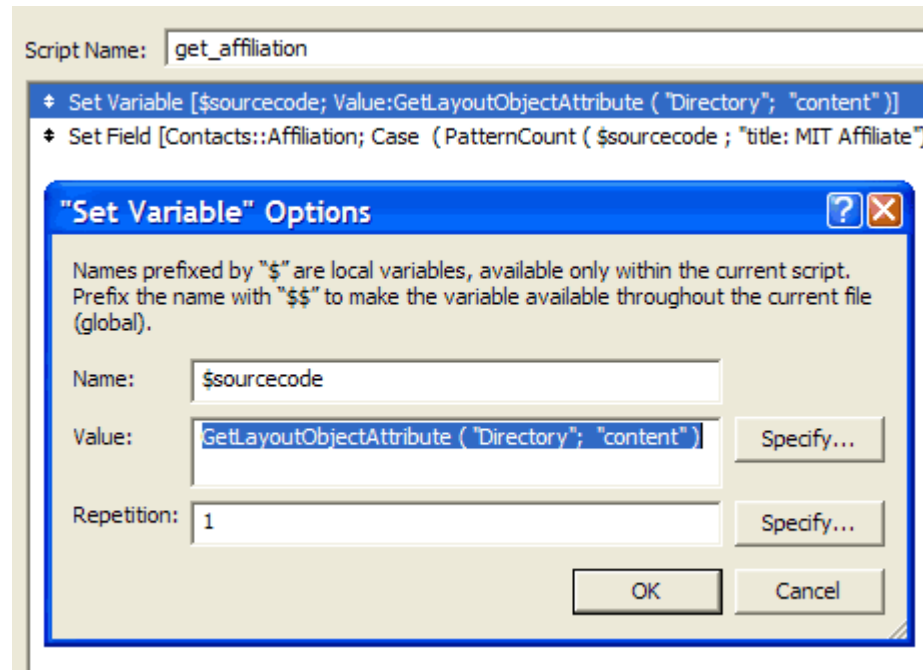


View the Web Page

- Go into Browse mode and look at what's displayed in your Web Viewer object
- Search for desired data that you wish to store in the appropriate FileMaker field.
- View the Source Code
 - Windows: Right-click in Web Viewer and select View Source
 - Mac:
 - Go back to your browser
 - Under the View menu, select View Source (Safari) or Page Source (Firefox)
- Note if the term appears more than once
 - This will determine your PatternCount
- Check a few possible records to scan for different possible values for the field

Build a script to scrape

- Script step to grab and store the source code:



Build a script to scrape (pt. 2)

- Script step to capture desired data:

† Set Variable [\$sourcecode; Value:GetLayoutObjectAttribute ("Directory"; "content")]
 † Set Field [Contacts::Affiliation; Case (PatternCount (\$sourcecode ; "title: MIT Affiliate") > 0; "Affiliate" ;

Specify Calculation ? X

This calculation will be evaluated based on context determined at runtime.

Contacts Address Type 1 Address Type 2 Affiliation City 1 City 2 Company Contact ID	Operators & / -- * ? - 0 +	= > < >= and	View: all functions by name Abs (number) Atan (number) Average (field {; field...}) Case (test1 {; result1 {; test2; resul... Ceiling (number) Choose (test ; result0 {; result1; re... Combination (setSize ; numberOfCh...
--	--	--------------------	---

```

Case
(
  PatternCount ( $sourcecode ; "title: MIT Affiliate") > 0; "Affiliate" ;
  PatternCount ( $sourcecode ; "year:") > 0; "Student" ;
  PatternCount ( $sourcecode ; "title:") > 0; "Staff" ;
  ""
)
      
```

Calculation result must be Text

Demo

Gotchas

- Best when used with a web site you can control, because:
 - If the url changes, it won't work
 - If any of the critical source code changes, you'll get bad or no data

What if?

- The value you're trying to scrape could be anything at all, like street addresses?
 - no set “value list” for your case statement

What you need

- FileMaker Pro v8.5 Advanced
 - Create a custom function
- Comfort with scanning web page source code
- Familiarity with creating scripts
- Familiarity with using functions and variables

View the Web Page Source Code

- Go into Browse mode and look at what's displayed in your Web Viewer object
- View the Source Code
 - Windows: Right-click in Web Viewer and select View Source
 - Mac:
 - Go back to your browser
 - Under the View menu, select View Source (Safari) or Page Source (Firefox)
- Search for desired data that you wish to store in the appropriate FileMaker field.
- Check a few possible records to scan for different possible values for the field
- Determine your bound markers

ParseData() Custom Function

```
// ParseData ( theText; theStartTag; theEndTag; theOccurance)
//
// Extract the text between two strings.
//
// Parameters:
// theText = the text to parse
// theStartTag = the string that comes before the text to extract
// theEndTag = the string that comes after the text to extract
// theOccurance = the instance of the text to extract
//
// Return Value:
// the instance of text found in theText between theStartTag and theEndTag based on theOccurance

Let ( [

theStartPos = Position ( theText ; theStartTag ; 1 ; theOccurance ) ;
theResult = Case (

// -----
// If theStartTag was not found, return an empty string.
theStartPos = 0 ; "" ;
// -----

// -----
// If theStartTag was found, get the string we are looking for.
theStartPos > 0 ;
Let ( [
theStartPos = theStartPos + Length ( theStartTag ) ;
theEndPos = Position ( theText ; theEndTag ; theStartPos ; 1 ) ;
theLengthToKeep = theEndPos - (theStartPos+1);
theResult = Middle ( theText ; theStartPos ; theLengthToKeep )
] ;
theResult
)
// -----

) // End case

];

theResult

)
```

The script step

Script Name:

- ◆ Set Variable [\$sourcecode; Value:GetLayoutObjectAttribute ("Directory"; "content")]
- ◆ If [Get (ScriptParameter) = "affiliation"
- ◆ Set Field [Contacts::Affiliation; Case (PatternCount (\$sourcecode ; "title: MIT Affiliate") > 0; "Affiliate"
- ◆ Else If [Get (ScriptParameter) = "map"]
- ◆ Set Field [Contacts::Street 1; If (PatternCount(\$sourcecode; "address: ">0); ParseData (\$sourcecode
- ◆ End If

Specify Calculation ? X

This calculation will be evaluated based on context determined at runtime.

<input type="text" value="Contacts"/>	Operators		View: <input type="text" value="all functions by name"/>
Address Type 1 Address Type 2 Affiliation City 1 City 2 Company Contact ID	& / * / - 0 +	= ≠ > < ≥ ≤ and	Abs (number) Atan (number) Average (field {; field... }) Case (test1 ; result1 {; test2 ; resul... Ceiling (number) Choose (test ; result0 {; result1 ; re... Combination (setSize ; numberOfCh...

```

{f ( PatternCount($sourcecode; "address: ">0);
ParseData ( $sourcecode ; "address: " ; 1 ; "department" );
"Web page data has changed. Contact your system administrator.")

```

Calculation result must be Text

Demo

Gotchas

- All the same gotchas apply as before
 - changes to URL
 - changes to source code
 - Test! Test! Test!

Resources

- The Web Viewer:

http://filemaker.custhelp.com/cgi-bin/filemaker.cfg/php/enduser/std_adp.php?p_faqid=6126

- ParseData() custom function:

– <http://www.briandunning.com/cf/559>

– (Thank you to Hal Gumpert!)