



JP Software

Take Command 9
Tutorial 4 - Working In The Internet World

Tutorial 4

Working In The Internet World

Take Command 9 has evolved to provide a bevy (yeah that's right a bevy) of features that allow you to work in an Internet centric world

Overview

Take Command 9.0 has the ability to:

- **Access & Manipulate Remote Sites** - You can get and put files in internet sites using several techniques including:
 - FTP (basic FTP)
 - TFTP (Trivial FTP)
 - FTPS (SSL FTP)
 - HTTP (basic Web access)
 - HTTPS (SSL HTTP)
- **Create Web Pages** - TC9 allows you to construct web pages and populate them with real time data from your system

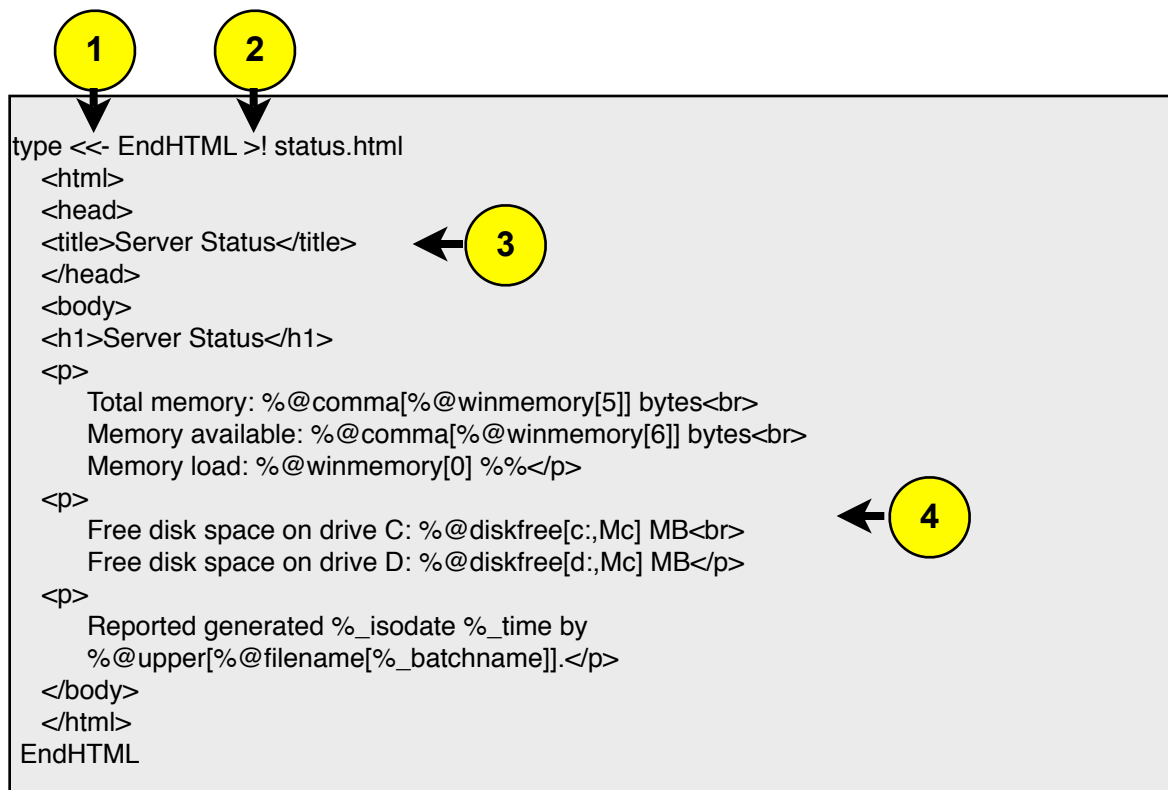
In this tutorial, we are going to show you a simple way to construct web pages with data from your computer and send them to a central website

Example 1 -- Create A Web Page

To create a web page, we use a very easy technique that was developed originally for Linux and has been implemented in Take Command.

The following script creates a web page (called status.html) and populates it with data about the status of your computer:

```
type <<- EndHTML >! status.html
<html>
<head>
<title>Server Status</title>
</head>
<body>
<h1>Server Status</h1>
<p>
  Total memory: %@comma[ %@winmemory[5]] bytes<br>
  Memory available: %@comma[ %@winmemory[6]] bytes<br>
  Memory load: %@winmemory[0] %%</p>
<p>
  Free disk space on drive C: %@diskfree[c:,Mc] MB<br>
  Free disk space on drive D: %@diskfree[d:,Mc] MB</p>
<p>
  Reported generated %_isodate %_time by
  %@upper[ %@filename[ %_batchname]].</p>
</body>
</html>
EndHTML
```



Here is what is happening:

1. <<- -- This creates a redirect that sends everything in the following lines up to EndHTML to the type command -- which creates a text output
2. >! -- says redirect the text output of the type command to a file called status.html. The ! after the redirection command (>) means that the system should overwrite any existing status.html file
3. **HTML Code** -- The next few lines are standard HTML code that sets up some static header text
4. **System Data** -- The next few lines gather data from the system. The parser will examine each line of code and do variable expansion. What this means is that if you precede text with a % sign, TC9 will assume that everything up to the next space is a variable or function and it will convert the variable or function to its actual value.

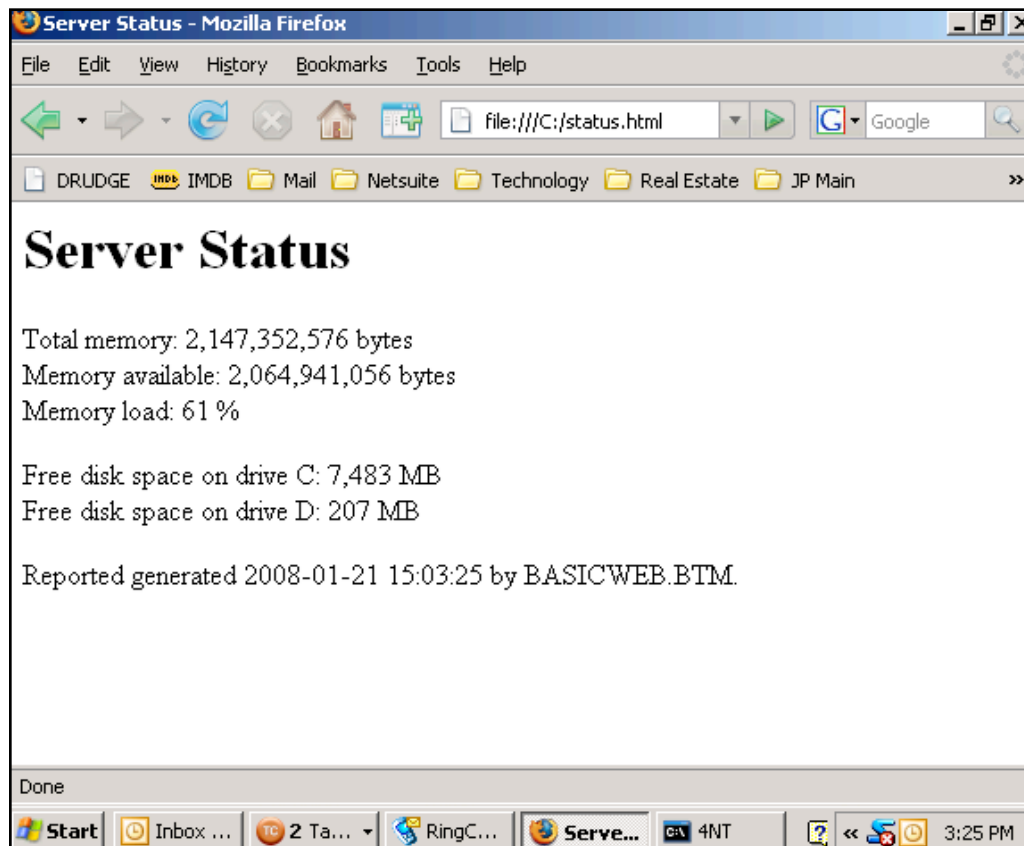
So, for example, `%@winmemory[5]` is evaluated as the actual amount of memory in the system. `%@winmemory[0]` is evaluated as the amount of memory (as a percentage) actually being used.

You can nest functions, so `%@comma[%@winmemory[5]]` will apply the comma function to the amount of memory returned, properly formatting it.

The box below shows what is in the status.html file after running the program.

```
<html>
<head>
<title>Server Status</title>
</head>
<body>
<h1>Server Status</h1>
<p>
Total memory: 2,147,352,576 bytes<br>
Memory available: 2,064,941,056 bytes<br>
Memory load: 61 %</p>
<p>
Free disk space on drive C: 7,483 MB<br>
Free disk space on drive D: 207 MB</p>
<p>
Reported generated 2008-01-21 15:03:25 by
BASICWEB.BTM.</p>
</body>
</html>
```

The following picture shows what the file looks like in a browser.

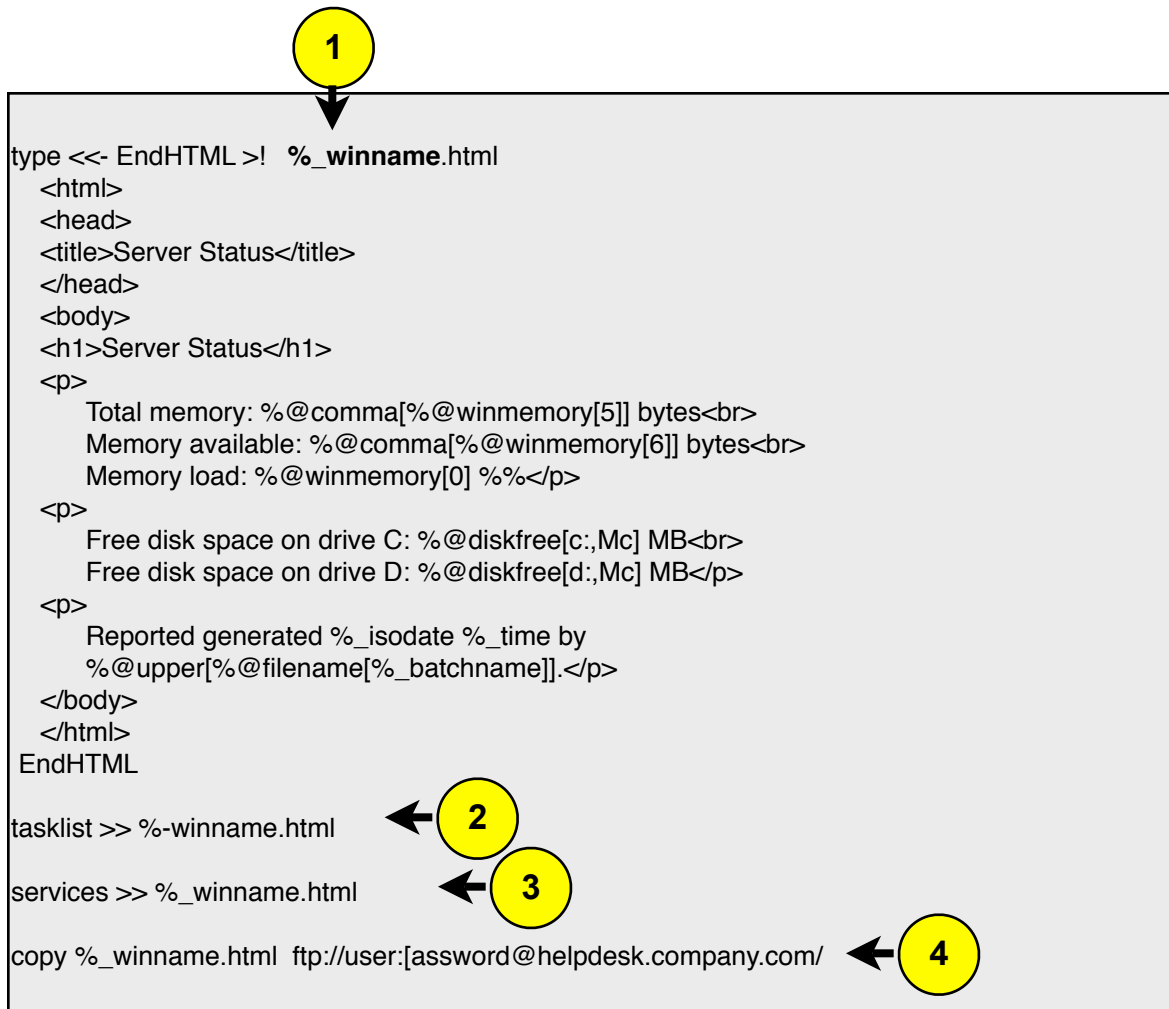


Pretty easy to do.

Example 2 -- Creating A Support System For Help Desks

If you have ever been part of a corporate help desk, you know that people often call in with issues about their computers, but you have no idea what is going on the computer. This example shows a simplified script you could put on all of the computers along with a desktop icon for users to press to execute the script. If a user has a problem this script will create a status web page and ftp: it to a central website so that the help desk group can get an idea what is going on in the computer.

We have expanded the code from the previous example



```
type <<- EndHTML >! %_winname.html
<html>
<head>
<title>Server Status</title>
</head>
<body>
<h1>Server Status</h1>
<p>
  Total memory: %@comma[ %@winmemory[5]] bytes<br>
  Memory available: %@comma[ %@winmemory[6]] bytes<br>
  Memory load: %@winmemory[0] %%</p>
<p>
  Free disk space on drive C: %@diskfree[c:,Mc] MB<br>
  Free disk space on drive D: %@diskfree[d:,Mc] MB</p>
<p>
  Reported generated %_isodate %_time by
  %@upper[ %@filename[ %_batchname]].</p>
</body>
</html>
EndHTML

tasklist >> %-winname.html
services >> %_winname.html

copy %_winname.html ftp://user:[assword@helpdesk.company.com/
```

In this example, we have added several features:

1. **%_winname** -- This is the name of the computer. It will create a unique file name
2. **Tasklist** -- The tasklist command outputs a list of all currently running processes. We are using >> to append the output of tasklist to the existing html file
3. **Services** - this is similar to the previous command, but in this case we are appending a list of system services to the html file
4. **Copy** -- the copy command is copying the html file to the company help desk website using ftp (with a user name and password used for security). Note that you can treat an ftp site as if it was a local hard disk - a great feature of Take Command.

This is a very simple example. Take Command includes hundreds of additional system variables and functions that can be used to gather status information.

In addition, if you understand Windows Management Interface (WMI), Take Command allows you to query anything known by WMI, which has almost all information about the status of the computer and activities going on in it.