

Desktop Info

by Glenn Delahoy



(C) Copyright 2005-2021
All rights reserved

Web Site: www.glenn.delahoy.com

Table of Contents

Description.....	5
License Agreement.....	5
Warranty.....	5
Technical Support.....	5
32 Bit Application.....	5
Windows Platforms.....	6
Application Priority.....	6
Process Access.....	6
Donate.....	6
General Usage.....	7
Context Menu.....	7
Command Line Options.....	8
Screen Saver.....	8
Windows Event Logs.....	9
Colors.....	9
Configuration File.....	11
Options.....	11
Screen Saver Section.....	17
Text Section.....	17
Pages Section.....	17
Functions Section.....	18
Items Section.....	18
Formatting And Transformation.....	20
Formatting Numbers.....	20
Transforming Numbers.....	21
Formatting Dates.....	22
Formatting Times.....	22
Formatting Booleans.....	23
Bit Mapped Numbers.....	23
Expressions.....	25
Functions.....	26
Languages.....	28
Character Sets.....	29
Right-To-Left Support.....	29
Language ID (LID).....	30
Options and Usage Topics.....	31
Key Words.....	31
Common Options.....	31
Item Colors.....	34
Thresholds.....	35
Charts.....	37
Moving To The New Charts.....	39
Legacy Charts.....	40
Data Logging.....	44
User Variables.....	45
Conditional Visibility.....	47

Left Column Text.....	47
Remote Monitoring.....	48
Item Reference.....	49
ALLIPADDRESS.....	49
BOOTTIME.....	50
CMD.....	50
COMMENT.....	54
CONTROL.....	54
CONTROL2.....	55
CORETEMP.....	56
CPU.....	56
CPUCOUNT.....	57
CPUTYPE.....	57
CPUUSAGE.....	57
DATADUMP.....	57
DATETIME.....	58
DIRECTX.....	58
DISPLAYCONTROLLER.....	58
DNSSERVER.....	59
DOMAIN.....	59
DOMAINCONTROLLER.....	59
DTI.....	59
ENVVAR.....	60
EVENTLOG.....	60
FILE.....	61
FILE2TEXT.....	61
FILECONTENTS.....	62
FILECSV.....	62
FILEEXIST.....	63
HTTPGET.....	64
HOST.....	64
IEVERSION.....	65
IF.....	65
LOADTIME.....	66
LOGICALDRIVES.....	67
NETCONNECTIONS.....	69
NETPACKETS.....	69
NETPACKETSRATE.....	69
NETWORKINTERFACES.....	70
OEMINFO.....	72
OSBUILD.....	72
PAGEFAULTS.....	73
PAGEFILE.....	73
PHYSICALDISKS.....	73
PHYSICALRAM.....	74
PROCESSCOUNT.....	74
PROCESSMEM.....	74
PROXY.....	75
RDS.....	75

RDSSESSIONS.....	76
REG.....	77
REGEXIST.....	78
REGISTRY.....	78
SERVICEPACK.....	79
SET.....	79
SNAPSHOT.....	80
TEXT.....	80
TIMEZONE.....	81
TOPPROCESSCPU.....	81
TOPPROCESSMEM.....	82
TOPPROCESSPF.....	83
UNREADMAIL.....	83
UPTIME.....	83
USER.....	84
VIRTUALMEMORY.....	84
WMI.....	84
Useful WMI Queries.....	87
Useful Windows Tools.....	90
References.....	92
Release Notes.....	93
Version v2.10.0.....	112

Description

This little application displays system information on your desktop. Looks like wallpaper but stays resident in memory and updates in real time. Perfect for quick identification and walk-by monitoring of enterprise, production or test farms or any computer you're responsible for. Everything is customisable (nearly).

License Agreement

This software is distributed free of charge. It may be used as many times as you like, for as long as you like, in a domestic or corporate environment. You may copy and distribute copies of this program provided that you keep all original documentation including this pdf file with copyright notice and disclaimer of warranty intact. You may not charge money or fees for the software product to anyone except to cover distribution costs.

Warranty

This program is provided "as is" without warranties of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the program is with you. Should the program prove defective, you assume the cost of all necessary servicing, repair or correction. In no event will the copyright holder be liable to you for damages, including any general, special, incidental or consequential damages arising out of the use or inability to use the program (including but not limited to loss of data or data being rendered inaccurate or losses sustained by you or third parties or a failure of the program to operate with any other programs).

Technical Support

No guarantees whatsoever are implied that technical support will be provided or that technical support, when provided, will be accurate. This software is basically unsupported and supplied on an as-is basis.

However there is a growing community at: www.glenn.delahoy.com/desktopinfo

32 Bit Application

This program is a 32 bit Windows application. It should run fine in any 64 bit environment with some restrictions. There will not be a native 64 bit version any time in the foreseeable future.

Windows Platforms

Desktop Info has been tested on the following Windows platforms:

Windows 2000 *

XP Pro (32 bit) version 2002 service pack 2

XP Pro (32 bit) version 2002 service pack 3

XP Pro (64 bit) version 2003 service pack 2

Vista (32 bit) business service pack 1

Server 2000 (32 bit) service pack 4

Server 2003 (64 bit) service pack 2

Server 2008 (32 bit) standard service pack 1

Server 2008 (64 bit) standard service pack 1

Server 2012 R2 (64 bit)

Server 2016

Server 2019

Windows 7 Professional 64 bit

Windows 8

Windows 10 Home/Pro 1709, 1803, 1809, 1903, 1909

WinPE 10 1809 x86

* The program basically works on Windows 2000 but some items won't work.

Application Priority

The application is set to *below normal* priority class as recommended by Microsoft which means that pretty much everything else will take priority but it won't wait until the cpu is idle.

Process Access

Some items may be denied access when collecting data depending on the access rights of the user running it on Vista, 7, 8, 10 and others. Most of the time this can be overcome by running Desktop Info with Administrator privileges.

Donate

Please show your appreciation for my hard work and long hours by making a donation. This will help keep me motivated and the project alive and healthy. Donations of \$100 or more will be acknowledged on the Special Donors page on the web site to be set up shortly and also here in the manual. You can use PayPal, your debit or credit card.

Go to www.glenn.delahoy.com/desktopinfo.

General Usage

Just run it. You can kill it via the right click context menu or from Task Manager. Open the desktopinfo.ini file or select Configuration from the right click context menu and adjust each item in the items section to control visibility, refresh times, colors and other properties. The display updates itself automatically when you save the ini file.

Desktop Info ships with several sample configuration files. The first time you run the application, a sample ini file will be copied to the working ini file. You are now free to modify it as needed using any text editor.

Context Menu

To get the context menu to appear in transparent mode, you have to right click on a visible pixel which may take a few tries. You can use the navigation buttons, a banner graphic or a bar chart. The context menu can be disabled by setting the 'contextmenu' item in the 'options' section to '0'.

Pages	Pages 1 to 9 are available depending on the configuration in the [pages] section.
Reload	This forces a reload of the configuration from the ini file.
Configuration	Opens the current configuration file in a text editor. When you save changes the configuration is reloaded.
Language	Opens the current language file in a text editor. When you save changes the configuration is reloaded.
Manual	Opens the manual with the default PDF reader.
Export	Export the display data using the selected template.
About	About box with version check and link to support page.
Quit	Closes the Desktop Info application.

See also the [\[text\] section](#) to change the text of these menu items.

The page titles can be set to any text or language using any of the following ways:

- The [\[pages\]](#) section in the main configuration file
- The [pages] section in the [language](#) configuration file
- The [PAGE_TITLE](#) key word in the [items] section of the main configuration file

The text for the other items on the context menu can be set using:

- The [\[text\] section](#) in the main configuration file
- The [\[text\] section](#) in the language configuration file

Individual options can be removed from the menu by setting the text to blank.

```
Menu-quit=
```

Export

The *Export* tool exports all active, non-hidden items using the selected template file. The *Export* menu is populated with files found in the *templates* sub-directory. Select a template from the menu to generate the export. The resulting file is automatically opened in the default viewer for the file type. From there you should be able to print it or save it to your preferred name and location. You can hide the Export menu by setting *menu-export* in the [\[text\] section](#) to blank.

The template file can be any file type, for example HTML, XML or RTF file. There are three special tags used in the template file to define where Desktop Info will insert the data.

<dti-items>	This defines the start of the item list. Everything between this and the </dti-items> close tag will be repeated for each display item.
<dti-text>	The left column text for the item.
<dti-display>	The right column text for the item.

See the example templates in the *templates* directory.

Command Line Options

The following command line options are available:

/f	Show as a normal window. This tells Desktop Info to appear like a regular application with a frame allowing you to move it to the background or foreground and drag it around like any other application.
/ini=	Specify the full path and file name to an alternative desktopinfo.ini file. Eg: desktopinfo.exe /ini=mynewinifile.ini Don't forget the equals sign.
/debug=	Enable debug mode. Specify the full path and file name to the debug log file.

If there is one option on the command line with no '/' prefix, it is assumed to be an ini file. This way you can drag an ini file onto the exe file to start Desktop Info using that ini file.

Screen Saver

Desktop Info can be used as a screen saver. Make a copy of DesktopInfo.exe and rename it to DesktopInfo.scr. Right click on this file and select Install. In the ini file, create a new section called [screen-saver] and add any of the options available in the [options] section. Some items that work in normal mode when run as Administrator may not work in screen saver mode due to reduced privileges.

Windows Event Logs

In addition to its own text file logging, you can make use of the Windows Event logs to monitor critical operations. Create an entry in the [options] section of the main ini file called "eventlog-level" and specify which logging events you want to be added to the event logs.

The available log levels are:

info	General application information
warn	When thresholds are triggered
error	Application or data errors

For example, to create an event log entry whenever a threshold is triggered or an error is raised:

```
eventlog-level=error warn
```

See also [Options](#).

Colors

Colors are specified in one of two ways:

BGR

A six character bgr (reverse rgb) hexadecimal number.

Think of this number as divided into three color elements, each element is a two digit number.

```
Color:BBGGRR
```

The first two digits represent the level of blue, the second represent the level of green, the third two represent the level of red.

RGB

A seven character rgb number. The first character is the hash symbol, '#', followed by a six character hexadecimal number.

Think of this number as divided into the prefix and three elements, each element is a two digit number.

```
Color:#RRGGBB
```

The hash prefix denotes rgb format, as opposed to the earlier bgr format, next is two digits that represent the level of red, two digits that represent the level of green and two digits that represent the level of blue.

Each color element has a range of 00-ff (or 0-255 in decimal). In bgr format, ff0000 is blue, 00ff00 is green, 0000ff is red. ffffff is white (all color elements at maximum), 000000 is black (all color elements are off). You can set each color element to any value between 00 and ff. Half way between fully off and fully on would be 80. If all three were set to half way, 808080, you would have a gray (grey) color.

If an item color is not defined it will take on the current default color. Use the key word [COLOR](#) to set the default color for a block of items.

Google Color Picker

Go to Google and type in “color picker”. This presents a simple color picker with a slider. Below the slider you’ll see the resulting hexadecimal (HEX) color number with the hash (‘#’) prefix. Select a color you like by clicking on the main color area and using the slider then highlight the complete hexadecimal number, including the hash prefix, and copy and paste this straight into your desktopinfo.ini file.



Configuration File

There is no configuration program. Options are set by modifying the ini file in a text editor such as Notepad. You can access the configuration from the right click context menu or directly using File Explorer. The ini file is divided into several sections. Each section contains a number of "key=value" entries.

Options

The following tables shows all options available in the [options] and [screen-saver] sections.

Size And Position

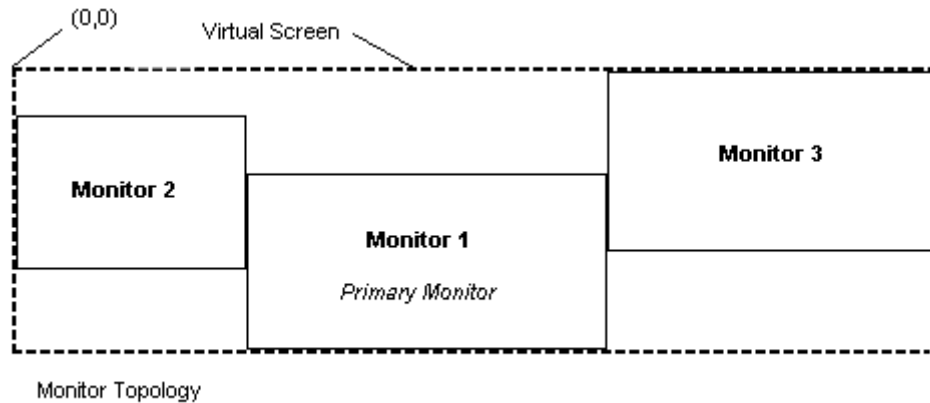
monitor-mode	0 = single monitor mode 1 = virtual screen mode
monitor-num	monitor number (set monitor-mode=0) 0 = primary monitor
top	Pixels down from the top
left	Pixels in from the left
bottom	Pixels up from the bottom
right	Pixels in from the right
width	Total width in pixels. If not specified then width is 400
centerv	Center vertically on the screen. 1 = centered 0 = not centered If centerv is specified, top and bottom are ignored.
centerh	Center horizontally on the screen. 1 = centered 0 = not centered If centerh is specified, left and right are ignored.
column1width	The width of column 1. -1 = automatic 0 = will effectively eliminate the column allowing the data column to occupy the whole space.

Desktop Info can view it's world from two different perspectives: single monitor mode and virtual screen mode.

Single Monitor Mode is the default mode. It's world consists of just the selected monitor. The origin (top=0 left=0), therefore, is the top left of the selected monitor. You cannot place DTI outside of the boundaries of the selected monitor regardless of the values you put in the top/left/bottom/right options. Using the monitor-num option, you can select which single monitor DTI lives in. 0 is the primary monitor, 1 is the secondary monitor etc.

So, to always have DTI on the primary monitor, set `monitor-mode=0` and `monitor-num=0` (this is the default). To force DTI to always be on the secondary monitor, set `monitor-mode=0` and `monitor-num=1`.

In Virtual Screen Mode, Desktop Info views it's world in terms of the Windows virtual screen. In this world the screen is an imaginary rectangle that encompasses all monitors connected to the system, regardless of their relative position. The origin (`top=0 left=0`) is the top left of this imaginary rectangle. In Virtual Screen Mode, the `monitor-num` option is not used.



Note: In version 1.11, the origin was always the primary monitor and therefore it was necessary to set values outside of the primary monitor. This is no longer the case, values now are always greater than or equal to zero and less than or equal to the width/height of either the single monitor or the virtual screen.

It is useful to note that if, like me, your laptop is in and out of the dock, DTI will automatically reset it's position according to the updated Windows display configuration. So, for example, I have a secondary monitor attached to the dock and configured to be above the laptop display. With DTI set to `monitor-mode=1 top=0 right=0`, it positions itself at the top right of the laptop display when it is out of the dock and the top right of the secondary monitor when it is in the dock.

Here are additional rules for the position options:

- if left is specified, it is left aligned
- if right is specified, it is right aligned
- if both are specified, it is left
- if neither then `right=0`

- if top is specified, it is top aligned
- if bottom is specified, it is bottom aligned
- if both are specified, it is top
- if neither then `top=0`

Interface

formcolor	background color (bgr)
transparency	background transparency. 0=opaque, 100=totally transparent
contextmenu	enable the right click context menu 0=disabled, 1=enabled
allowdrag	enables the ability to drag the form. If the /f option is used the form is always draggable. 0=disabled, 1=enabled
nav	enable the navigation buttons 0=disabled, 1=visible at the bottom of the display (default) 2=visible at the top of the display, below the logo
nav-align	horizontally align the navigation buttons -1=align left, 0=align center (default), 1=align right
nav-color	navigation buttons color (bgr)
language	file name containing language override settings. See Language File section for more information.

The optional navigational buttons may be enabled using the "nav" option. If *nav=1*, the buttons appear at the bottom of the display below the last item. If *nav=2*, they appear at the top of the display or below the logo if it is enabled. They may be aligned left, center or right using the "nav-align" option.

You can adjust the DTI transparency using the 'transparency' option. A value of 100 is totally transparent, a value of 0 is opaque. You can also set the background color of a partially transparent form using the 'formcolor' option.

There is just one caveat to the colors: because Windows requires that I select a color to act as the transparent color, I chose black. This means if you set any of the items to be black color, they will become transparent. You should instead use '000001' or some other color that is close enough to black.

Font

font-face	the name of the default font face, may be overridden by an item or key word within the <i>[items]</i> section
font-size	default font size, may be overridden by an item, default is 8, minimum is 6, may be overridden by an item or key word within the <i>[items]</i> section
font-charset	the character set for the font
font-rtl	Use right-to-left language rules to display the text. 0=left-to-right (default), 1=right-to-left
font-quality	set the anti-aliasing level 0=system default, 1=draft, 2=proof, 3=nonantialiased 4=antialiased (default), 5=cleartype
line-spacing	the number of pixels between each row (default=1)

The *font-quality* option effectively sets the level of anti-aliasing. According to Microsoft:

The FontQuality Enumeration specifies how closely the attributes of the logical font match those of the physical font when rendering text.

DEFAULT_QUALITY	= 0,
DRAFT_QUALITY	= 1,
PROOF_QUALITY	= 2,
NONANTIALIASED_QUALITY	= 3,
ANTIALIASED_QUALITY	= 4,
CLEARTYPE_QUALITY	= 5

DEFAULT_QUALITY: Specifies that the character quality of the font does not matter, so DRAFT_QUALITY can be used.

DRAFT_QUALITY: Specifies that the character quality of the font is less important than the matching of logical attributes. For rasterized fonts, scaling SHOULD be enabled, which means that more font sizes are available.

PROOF_QUALITY: Specifies that the character quality of the font is more important than the matching of logical attributes. For rasterized fonts, scaling SHOULD be disabled, and the font closest in size SHOULD be chosen.

NONANTIALIASED_QUALITY: Specifies that anti-aliasing SHOULD NOT be used when rendering text.

ANTIALIASED_QUALITY: Specifies that anti-aliasing SHOULD be used when rendering text, if the font supports it.

CLEARTYPE_QUALITY: Specifies that ClearType anti-aliasing SHOULD be used when rendering text, if the font supports it.

Fonts that do not support ClearType anti-aliasing include type 1 fonts, PostScript fonts, OpenType fonts without TrueType outlines, rasterized fonts, vector fonts, and device fonts.

The default font quality is set to 4, which is the best anti-aliased mode other than clear type quality. If you want clear type fonts, add “*font-quality=5*” to the options section.

Banner Graphic

Optional graphic at the top of the display. The image can be jpg, tif, png, gif, bmp, wmf, emf format.

logo	graphic file to display at the top of the Desktop Info display logo=<filename>
logo-width	stretch the image to this width 0=maintain the width of the original image (default) -1=stretch to the width of the form
logo-height	stretch the image to this height 0=maintain the height of the original image (default) -1=stretch to the height of the form (why?)
logo-align	horizontal positioning relative to the form width -1=align left 0=align centre (default) 1=align right

Application Logging

log	Write application logging information to the specified log file.
log-level	info=general information messages error=application or data errors warn=thresholds debug=for tracking down application issues data=displayed data output debugonerror=start debug logging for a short time after an error is raised
eventlog-level	Write entries to the Windows Application event log info=general information messages error=application or data errors warn=thresholds

DesktopInfo64

Tool used to retrieve 64 bit counters and other metrics not available to a 32 bit application.

desktopinfo64	Enable the 64 bit tool 0=disabled (default), 1=enable
---------------	--

Why do we need 64 bit counters? Desktop Info is a 32 bit application. In nearly every case this is fine. In a few cases this has been a barrier to retrieving accurate counters. DesktopInfo64 is a small 64 bit application that performs these tasks in a separate process and passes the results back to the main application.

The items that this tool handles are [TOPPROCESSMEM](#), [PROCESSMEM](#) and [PROCESSCOUNT](#). In the case of the first two, because a 32 bit number has a maximum value of 4GB and some big processes can consume more than 4GB of memory. Using the 64 bit tool enables values greater than 4GB to be retrieved from the counters.

In the case of [PROCESSMEM](#) and [PROCESSCOUNT](#), Windows does not allow 32 bit applications to access process information for 64 bit applications so the 64 bit tool is used to retrieve some process counters including commit size and thread count.

When enabled, desktopinfo64 is started automatically when Desktop Info starts and shut down when Desktop Info shuts down. It will inherit access permissions so if Desktop Info is run as Admin, then desktopinfo64 is also run as admin.

Remote Monitoring

Desktop Info can monitor other computers. It does this by running a small internal http server, allowing another instance of Desktop Info to connect and retrieve raw data. This raw data is stored on the client instance as if the client had retrieved it locally allowing full display formatting of the data.

So far I've only tested this within the LAN but in theory it should work over the internet or across network boundaries given appropriate port forwarding etc. This is a potential security issue so use with caution. Your anti-virus and/or firewall may react when the server is switched on. To monitor items that are not on the current page, enable the *background-interval* option so that there is data to serve up. In this way, you could in theory keep DTI completely invisible while still monitoring remotely.

To start the internal http server, set the following options in the *[options]* section of the ini file.

server-ip	the ip associated with a network interface card
server-port	the port on the above ip

The server-ip must be an IP address associated with a network interface on the host. If it is not defined or invalid, the server won't start. This means that, in the long term, the host should have a reserved or static ip.

The server-port must be open on the host firewall. If it is 0 or not defined, the server won't start.

You can test the server by opening your browser and navigating to the server in question:

```
http://hostname:8192/data?id=cpu1
```

You need to specify the http protocol, there is no SSL option. Insert your host name and port number and set the id to any unique item id. If everything is working correctly, you will see an xml snippet where the root node is *raw_values* followed by 0 or more rows of data.

```
<raw_values>
  <row>
    <data type="3">3</data>
    <data type="3">1</data>
    <data type="6">0</data>
  </row>
</raw_values>
```

Other

inimontortime	how often to check for desktopinfo.ini changes
text-editor	Full path and file name to your preferred text editor. Called when you open the configuration and language files from the context menu.
align	Set the left/right text alignment for both columns 0=left aligned (default), 1=right aligned
text-align	Set the alignment of the left column text. Overrides the ALIGN setting. 0=left aligned (default), 1=right aligned
initial-page	set which page is initially selected when DTI starts
auto-home	seconds of inactivity until DTI returns to the home page 0=disabled (default)
ignore-setting-change	Suppress some redraws by ignoring the Windows WM_SETTINGCHANGE message triggered by events such as desktop background color changes and remote desktop session connects.

0=don't ignore (default), 1=ignore

Text Alignment

Text alignment can be set at the global *[options]* level and at the item level. If you set it once in the *[options]* section all items will follow. You can additionally set individual items to override the global settings. The rules are:

If no options are specified, all text is left aligned.

If global ALIGN is set, alignment is set for all items.

If global TEXT-ALIGN is set, left column text alignment is set for all items. Overrides the ALIGN option.

If the item ALIGN option is set, the global ALIGN option is overridden for that item only.

If the item TEXT-ALIGN option is set, the global and/or item ALIGN option is overridden and the global TEXT-ALIGN option is overridden for that item only.

Screen Saver Section

This section is called *[screen-saver]* and is identical to the *[options]* section above but is used in screen saver mode. If this section is not found when running in screen saver mode then the normal *[options]* section is used.

The *initial-page* option can be useful here to set a different page to show on the screen saver to what is initially shown in normal mode.

It is also useful to use relative font sizing for your items so that scaling up the font size for the screen saver is a simple case of raising the default *font-size* in the *[screen-saver]* options.

Text Section

The following table shows all options available in the *[text]* section.

For the menu items, if the text is set to nothing, the item is removed from the menu.

noresults	when an item returns no results
error	on screen and in log file when an error occurs
null-result	when a wmi return value is null
unknown	when a logical drive has no file system or volume
menu-reload	in the context menu for the "Reload" item
menu-configuration	in the context menu for the "Configuration" item
menu-language	in the context menu for the "Language" item
menu-manual	in the context menu for the "Manual" item

menu-export	in the context menu for the Export parent item
menu-about	in the context menu for the “About” item
menu-quit	in the context menu for the “Quit” item
nav-previous	in the hover hint for the previous page button
nav-home	in the hover hint for the home page button
nav-next	in the hover hint for the next page button

Pages Section

You can configure up to 9 pages of data to be available for viewing from the right click context menu and the navigation buttons. This is a neat way of grouping information together, providing a summary on the first page and more detailed info on other pages. This section is optional. If it is not present then only the first page, titled ‘Home Page’, is available.

The pages section allows you to configure the page titles. Give each page whatever title you want. A blank title will disable that page. Page 1 cannot be disabled.

```
[pages]
1=Home Page
2=Windows / Hardware
3=
4=
5=
6=
7=
8=
9=
```

Use the PAGE key word in the [items] section to set the page for the following items.

By default, DTI only collects data for items on the currently visible page. When you change the page the data will begin collecting for the items on the new page. You can force selected items to collect data even when not visible by adding the [*background-interval*](#) option.

Functions Section

This section allows you to declare any number of functions to be used in item display templates. This section is optional.

The general format is:

```
[functions]
name=expression
```

Where *name* is the function name you will reference from within an item display template and *expression* is the arithmetical expression to be evaluated. The expression will include replaceable parameters to represent the arguments passed in to the function. So %1 is the first argument, %2 is

the second argument, etc. Before the expression is evaluated, these replaceable parameters are replaced with the actual values from the item returned values.

Here are two example functions that convert Kelvin to Celsius and Kelvin to Fahrenheit.

```
[functions]
KtoC=%1-273.15
KtoF=(%1-273.15)*9/5+32
```

Where %1 is the temperature value passed in from the calling item display template.

See the discussion in the [Functions](#) section for more information.

Items Section

The items section in the ini file sets the order and state of each item on the screen. Each item consists of an ID followed by a comma delimited list of key:value options.

A single item can be split over several lines by ending a line with the backslash character. The lines are joined back together as the config is loaded. For example:

```
COMMENT=color:%white%,style:w,font-face:Ink Free,\
font-size:160%,text:Desktop Info by Glenn Delahoy
```

You can indent the lines with spaces for readability. If the line is split in the middle of the display template, any spaces used to indent will be included in the display output.

See the [Item Options](#) section for more information.

The complete list of available items is in the [Item Reference](#) section.

Formatting And Transformation

Formatting Numbers

Any item that displays a number can be formatted in the 'display' template property. You specify the replaceable parameter for the value you want (refer to the item reference) and follow it with the number format option. The general form of the number format is:

```
%1[w.pt]
```

This is the replaceable parameter representing the value you want to display immediately followed by exactly four characters inside square brackets. The first character 'w' is a single digit that defines the maximum width of the number. The second character is a dot. The third character 'p' is a single digit that defines the precision of the number. That is, the number of places after the decimal point. The fourth character 't' defines the type of number.

The four general number types are:

d	decimal	w = minimum width left padded with zeros, p=not used
f	float	w = minimum width left padded with spaces, p=decimal places
n	float	same as 'f', commas inserted for thousands
x	hexadecimal	w = minimum width left padded with zeros, p=not used

In this case, decimal means integer, a whole number, float means a number that may not be a whole number. Hexadecimal can only be used on integer type numbers.

Example 1:

```
display:%1[3.0d]
```

This displays the first value with no decimal place, a minimum 3 characters wide, left padded with zeros. Eg "005" or "016" or "000" or "83738"

Example 2:

```
display:%2[5.0d]
```

This displays the second value with no decimal place, a minimum 5 characters wide, left padded with zeros. Eg "00040" or "00597" or "00000"

Example 3:

```
display:%3[4.1f]
```

This displays the third value with 1 decimal place, a minimum 4 characters wide, left padded with spaces. Eg "12.5" or " 9.0" or " 0.1"

Example 4:

```
display:%1[7.0n]
```

This displays the first value with no decimal places, a minimum 7 characters wide, commas inserted for thousands, left padded with spaces. Eg “123,456” or “ 3,654” or “ 67”

Transforming Numbers

Sometimes it's useful to change a number from thousands of things to kilothings or megathings or gigathings. This is usually bytes but it doesn't have to be. Formatting types are available to help transform these values.

These also follow the rules for the 'f' number type. Width excludes the units in the case of 'b' or 'B'.

Note: There is a difference between binary bytes and decimal bytes. Binary bytes are powers of 2. So a kilobyte is 1024 bytes. Decimal bytes are powers of 10. So a kilobyte is 1000 bytes. Make sure you choose the correct transform option for the data. For example, most hard drive manufacturers quote their drive size in decimal bytes, Windows quotes memory in binary bytes.

See also: <https://en.wikipedia.org/wiki/Kilobyte>

In the following table, binary transformations use the lower case number type, decimal transformations use the upper case number type.

k K	convert to kilo	w = minimum width left padded with spaces, p=decimal places
m M	convert to mega	w = minimum width left padded with spaces, p=decimal places
g G	convert to giga	w = minimum width left padded with spaces, p=decimal places
t T	convert to tera	w = minimum width left padded with spaces, p=decimal places
b B	best fit convert	w = minimum width left padded with spaces, p=decimal places, append unit

Example 1:

```
%3[1.0k]
```

Displays the third value, binary converted to kilothings, no decimal places, a minimum of 1 character wide. eg "4" or "0" or "4567"

Example 2:

```
%3[3.1M]
```

Displays the third value, decimal converted to megathings, 1 decimal place, a minimum of 3 characters wide. eg "4.1" or "12.0" or "1483.6" or "0.5".

Example 3:

```
%3[3.1b]
```

Displays the third value, binary converted using best fit, 1 decimal place, a minimum of 3 characters wide, unit appended. eg "6.1Ki" or "6.5Mi" or "1.1Gi".

Desktop Info - Manual

If that was a decimal transform:

```
%3[3.1B]
```

Displays the third value, decimal converted using best fit, 1 decimal place, a minimum of 3 characters wide, unit appended. eg "6.1K" or "6.5M" or "1.1G".

You add your own 'thing' after the number. Example:

```
%3[3.1b]B
```

Displays the third value, binary converted, "6.1KiB".

or

```
%3[3.1B]B
```

Displays the third value, decimal converted, "6.1KB".

Formatting Dates

The following can be used to help format dates:

[ddd]	short day name
[dddd]	long day name
[mmm]	short month name
[mmmm]	long month name
[yy]	last two digits of the year
[yyyy]	four digit year

Example 1:

```
%1[ddd]
```

The first value is the day of the week. It can be displayed as a normal number or formatted to show the short day name. eg "Mon" "Tue".

Example 2:

```
%1[dddd]
```

The day of the week can be displayed as a long day name. eg "Monday" "Tuesday".

Example 3:

```
%1[dddd] %2 %3[mmmm] %4[yyyy]
```

This will display the first four values of DATETIME. eg "Thursday 6 September 2018".

Example 4:

```
%1[ddd] %2 %3[mmm] %4[yy]
```

This will display DATETIME as "Thu 6 Sep 18".

Formatting Times

If you want a basic 24 hour display, the regular number format would work:

```
%5[1.0d]:%6[2.0d]:%7[2.0d]
```

Will display something like "9:24:45" or "13:15:01".

To format a 12 hour display there are a few additional format options:

[1.0a]	convert the hour to 12 hour, ie any value from 13 to 24 is reduced by 12
[2.0p]	show 'am' or 'pm' depending on the hour in the value
[2.0P]	show 'AM' or 'PM' depending on the hour in the value

Example 1:

```
%5[1.0a]:%6[2.0d]:%7[2.0d] %5[2.0P]
```

This will display DATETIME as "9:24:45 AM" or "1:15:01 PM". Notice that the last parameter is the hour (%5) but in this case is being used to decide how to display the AM or PM.

Example 2:

```
%5[1.0a]:%6[2.0d]:%7[2.0d] %5[2.0p]
```

This will display DATETIME as "9:24:45 am" or "1:15:01 pm".

Formatting Booleans

Boolean values can have 1 of only 2 possible values: 1 or 0, true or false, yes or no, on or off etc. In WMI land, booleans tend to be displayed as True or False. Sometimes it makes sense to display these values using normal everyday language. To convert boolean values into words use the following format:

```
[b:true:false]
```

In the place of the word 'true', you put whatever text you want. The same with the word 'false'. The correct text will be displayed based on the value of the boolean number.

In the following example, Win32_ComputerSystem has two properties relating to daylight savings. These can be displayed meaningfully thus:

Example 1:

```
display:Enabled: %EnableDaylightSavingsTime%[b:Yes:No]\, In Effect: %DaylightInEffect%[b:Yes:No]
```

This will display something like "Enabled: Yes, In Effect: No".

Bit Mapped Numbers

There are some WMI classes that return multiple results in a single number. We call these "bit mapped" numbers. For example, the AntiVirusProduct class in the root\SecurityCenter2 namespace has a property called ProductState that is a single integer value. Each binary 'bit' in that integer

Desktop Info - Manual

represents some information like 'enabled' and 'up to date'. We can extract this information by performing a bit-wise AND operation.

Desktop Info achieves this using the following display format:

```
[bit:value:true:false]
```

This is very similar to the boolean format except we want to know if a given bit is on in the returned property value. The 'value' is the decimal value of the bit you want to examine. In the place of the word 'true', you put whatever text you want. The same with the word 'false'. Desktop Info performs the bit-wise AND operation with the 'value' and the returned property value. If the result is not zero, the 'true' text is displayed, if the result is zero, the 'false' text is displayed.

In the case of the AntiVirusProduct example, the WMI query will look like this:

```
WMI=text:AV,namespace:root\SecurityCenter2,  
query:AntiVirusProduct,display:%productState%
```

This will show a number something like 397312 or 393216. This is not very useful. Now we add the display format:

```
... display:%productState%[bit:4096:Enabled:Disabled]
```

The display will now show text depending on whether the given bit is on or off:

```
AV                Enabled  
or
```

```
AV                Disabled
```

There is another bit field in that number we can make use of which is the 'up to date' field. The decimal value for this field is 393216. The display format will look like:

```
display:%productState%[bit:393216:up to date:not up to date]
```

The display will show text depending on whether that bit is on or off (it's actually two bits but that matters not):

```
AV                up to date  
or
```

```
AV                not up to date
```

Finally, we can combine these two in the display format thus:

```
display:%productState%[bit:4096:Enabled:Disabled] and %productState%  
[bit:393216:up to date:not up to date]
```

This will result in a display something like:

AV Enabled and up to date
or

AV Enabled and not up to date
or

AV Disabled and not up to date

Expressions

You can add arithmetical expressions to a display template in order to massage item return values or calculate additional information for display. For example, to convert a temperature return value from Kelvin to Celsius or Fahrenheit, to display a percent value calculated from two item return values. If there is any part of the expression that cannot be evaluated (ie illegal characters), an error will result.

The general format for an expression is:

```
{{expression}}
```

In other words, it's just enclosed in double braces. A trivial example is:

```
display: {{1+2}}
```

will display the value of '3'.

Normal arithmetical syntax and rules apply such as brackets and priority of evaluation.

```
display: {{1+2*3}}
```

will display a result of '7', whereas:

```
display: {{(1+2)*3}}
```

will display a result of '9'.

Use the item return values as you would elsewhere in the display template:

```
display: {{%1+%2}}
```

Will add the first two return values.

For WMI queries, use the wmi properties as you would elsewhere in the display template:

```
display: {{%temperature%-273.15}} C
```

You can add a format argument. The general format is:

```
{{format:expression}}
```

This will format the result using the standard number [formatting rules](#). For example:

```
display: {{1.1f:1+2}}
```

will result in the display, '3.0'.

A real world example is a WMI query to get the CPU temperature in Celsius. This particular WMI class returns temperatures in tenths of degrees Kelvin. The display template contains an expression to convert that number into degrees Celsius.

```
WMI=interval:10,text:Cpu Temp,namespace:root\wmi,  
query:MSAcpi_ThermalZoneTemperature, display:{{1.0d:  
%CurrentTemperature%/10-273.15}}Celsius
```

Here's a similar example to return the temperature in Fahrenheit. Note the extra pair of brackets in the formula.

```
WMI=interval:10,text:Cpu Temp,namespace:root\wmi,  
query:MSAcpi_ThermalZoneTemperature, display:{{1.0d:  
(%CurrentTemperature%/10-273.15)*9/5+32}}Fahrenheit
```

The following mathematical and trigonometric functions are available:

```
Abs, ArcCos, ArcCosD, ArcCosh, ArcSin, ArcSinD, ArcSinH, ArcTan,  
ArcTanD, ArcTanH, Ceil, Cos, CosD, Cosh, Cotan, DegToRad, Exp, Floor,  
Frac, GradToRad, Log10, Log2, Ln, Max, Min, Mod, Power, RadToDeg,  
RadToGrad, Round, Sin, SinD, SinH, Sqr, Tan, TanD, TanH, Trunc
```

All functions are called as Func(x). For example, Abs(100), Round(%2). The Min, Max, Mod and Power functions require two arguments separated by a comma, Min(10,20). Because we are inside the display template, we need to escape the comma, Min(10\,20).

```
display: {{min(%1\,%2)}} {{mod(10\,3)}}
```

Functions

This is a variation on the expression. (Not to be confused with the above mathematical / trigonometric functions.) The advantage of using functions over expressions is you only need to declare an arithmetic expression once and then reference it from any item as needed. Otherwise the two perform the same purpose.

The general format for calling a function is:

```
{{function:format:arg1[:arg2:...]}}
```

Where *function* is the name of a function defined in the [\[functions\]](#) section of the ini file, *format* is a standard format as defined by the standard number [formatting rules](#), followed by one or more arguments to be passed in to the function, each separated by a colon. Each argument is essentially

an expression that may comprise a mixture of item return values and arithmetical expressions. However, the most common case will be where each argument is a return value or WMI property name to be passed in to the function as arguments.

See the [\[functions\]](#) section for more information.

Working Example

Create a function to convert tenths of degrees Kelvin to Fahrenheit and use it to display WMI results.

```
[functions]
KtoF=(%1/10-273.15)*9/5+32

[items]
WMI=interval:10,text:Cpu Temp,namespace:root\wmi,
query:MSAcpi_ThermalZoneTemperature, display:{{KtoF:1.0d:
%CurrentTemperature%}}
```

In the [functions] section, KtoF is a function that contains the formula required for the conversion. %1 represents the value that will be passed to the function.

In the [items] section, the WMI query display template contains the call to the KtoF function. It passes in the temperature property and defines a number format for display.

Languages

The main ini file now fully supports multi-byte so if you want to use a single non-English language you can modify the ini file directly. All of the [font options](#) in the [options] section apply and you must save the ini file with UTF-8 encoding. This section deals with the situation where you want several language options available to override the default language in the main ini file.

The language file allows you to override text and display options set in the desktopinfo.ini file. It essentially mirrors the main ini file with the exception of the actual item configuration. You don't have to supply every configuration option again in the language file, just specify the ones you want to be different from the main ini file.

All options in the [options] section can be overridden as required. If you override the [pages] section you should specify all pages even if they are blank. The menu items are not multi-byte compatible yet, your mileage will vary.

In the language file, the [items] section contains a list of 'text' values that will override the item text in the main ini file. That is, the actual *text* option in each item. Add just the item entries you want to override. Each line is a name=value pair where the name matches either the item id or the item lid in the main ini file and the value is any text you want to use on the title display for that item.

Desktop Info is able to display multi-byte and right-to-left languages for the most part. If your language is a multi-byte language, it's important the language text file is UTF-8 encoded so that the text is correctly stored and read back and the font options are correctly specified.

To successfully display multi-byte languages you need the following things:

- Specify a language file in the desktopinfo.ini file
- The language file must be UTF-8 encoded
- In the language file, specify the "font-face" option with a font capable of displaying your language
- In the language file, specify the "font-charset" option with a character set within the font
- In the language file, specify the "font-rtl" option if your language is right-to-left

Desktop Info comes with a number of sample language files found in the "sample-languages" directory in various degrees of translation completion. If you decide to use one of the sample language files, you should copy it to a new location first to avoid it being overwritten by the next new version. Some of the sample language files are supplied by volunteers for which I am grateful while some are constructed from Google Translate so I don't expect they will be wonderful translations. If you see any errors please let me know so I can correct them for future releases.

Character Sets

If you wish to display a language that is different to your current Windows locale, you may need to set the "font-charset" option in your language file so the correct character set is used for display. Latin based languages normally use the Ansi character set.

For example, to set the character set to display simplified Chinese:

```
font-charset=134
```

This is the list of available character sets from which you can choose.

0	ANSI_CHARSET	ANSI characters
1	DEFAULT_CHARSET	Font is chosen based solely on Name and Size. If the described font is not available on the system, Windows will substitute another font.
2	SYMBOL_CHARSET	Standard symbol set.
77	MAC_CHARSET	Macintosh characters.
128	SHIFTJIS_CHARSET	Japanese shift-JIS characters.
129	HANGEUL_CHARSET	Korean characters (Wansung).
130	JOHAB_CHARSET	Korean characters (Johab).
134	GB2312_CHARSET	Simplified Chinese characters (mainland china).
136	CHINESEBIG5_CHARSET	Traditional Chinese characters (Taiwanese).
161	GREEK_CHARSET	Greek characters.
162	TURKISH_CHARSET	Turkish characters.
163	VIETNAMESE_CHARSET	Vietnamese characters.
177	HEBREW_CHARSET	Hebrew characters.
178	ARABIC_CHARSET	Arabic characters.
186	BALTIC_CHARSET	Baltic characters.
204	RUSSIAN_CHARSET	Cyrillic characters.
222	THAI_CHARSET	Thai characters.
238	EASTEUROPE_CHARSET	Includes diacritical marks for eastern European countries.
255	OEM_CHARSET	Depends on the code page of the operating system.

Right-To-Left Support

Desktop Info supports Right-To-Left languages. In addition to the "font-charset" option above, add the following option to your ini file or language file:

```
font-rtl=1
```

Language ID (LID)

If you want to override text for an item that does not have a unique ID, you will need to specify a 'lid' for the item. That is, where the ID appears more than once in the main ini file such as COMMENT, WMI etc. If the ID is unique then that can be used in the language file.

So, to override an item you know to be unique, such as CPUTEMP, just add an entry to the language file [items] section with that key and the text you want.

```
CPUTEMP=Température du Cpu
```

If the item ID is not unique, such as COMMENT, add the "lid" option to the item in the main ini file:

```
COMMENT=text:Desktop Info,lid:banner
```

Now add an entry with that key to the language file [items] section:

```
banner=Infos sur le bureau
```

You can have multiple instances of any item, for example you might put DATETIME on each page. In this case you don't need to have a unique language id because the text you define in the language file for the DATETIME item would apply to all instances of the item.

You should ensure the language file is UTF-8 encoded for multi-byte compatibility.

To enable the new language add the 'language' entry to the [options] of desktopinfo.ini pointing to your language file.

```
language=french.ini
```

Options and Usage Topics

Key Words

The following keys may be used anywhere in the [items] section. When encountered, all following items will assume these values until a new one is set. An individual item may continue to set it's own value independently of these default values. Key words may be used any number of times to set new item defaults.

In versions prior to v2.2, when an item color is set, all following items would assume this color. This is no longer the case, an individual item color will now simply override the current default for itself without changing the default.

PAGE	Current default item page number
PAGE-TITLE	Menu title of the current default page
FONT-FACE	Item font
FONT-SIZE	Item font size
COLOR	Item color both columns or right column
TEXT-COLOR	Item color left column

Example:

```
[items]
PAGE=2
PAGE-TITLE=Dates and Times
COLOR=%Silver%
COMMENT=style:iwb,text:Dates / Times
TIMEZONE=interval:3600
```

Also note that the [\[pages\] section](#) and the [active item option](#) will be removed in a future version in favour of the PAGE and PAGE-TITLE key words noted here.

Common Options

The following table lists options common to all items. In addition, each item will define it's own specific options detailed in the [item reference](#).

<u>key</u>	<u>value</u>
active	0=not active, 1 or more is active on that page number
hidden	0=not hidden (default) 1=item is active (collecting data) but will not display
hide-no-result	0=never hide (default) 1=hide the item if there is no result (no data)
interval	The refresh interval in seconds 0=never refresh after the first collection
background-interval	The refresh interval in seconds when the item is not on the currently visible page. This allows you to collect data for any item that is not visible and update charts and write csv

	data files in the background.
lid	Language id. This is used to look up the text in the language file that is used for the left column.
color	A bgr or rgb value as described in the Colors section. This overrides the current default COLOR value. If <i>text-color</i> is not specified then this controls the color for both columns.
text-color	Overrides the current default TEXT-COLOR value and specifies the color of the text in the left column. If not specified then the left column color is specified by the <i>color</i> option.
style	Font style b=bold, i=italic, u=underline, w=full width underline
csv	output data to a csv file
csvdatatype	0=raw data, 1=display data
font-face	overrides the default font face for an item
font-size	overrides the default font size for an item, this can be an absolute font size or a percent of the default font size specified in the [options] or [screen-saver] sections, see below for a discussion.
font-rtl	Use right-to-left language rules to display the text. Overrides the global setting. 0=left-to-right (default), 1=right-to-left
threshold1..9	up to 9 thresholds may be defined for each item
chart	Display a bar or line chart as described in the Charts section.
bar-colors	Bar chart colors in bgr or rgb form as described in the Bar Chart Colors section.
set	Save the display output or raw data to the given user variable. See below for more information.
display	template for the item right column display
text	Set your own item title text in the left column. Appears once on the first display row of an item. See discussion here .
align	Align both columns text. Overrides the global setting. 0=left align (default), 1=right align
text-align	Align the left column title text. Overrides the global setting and any ALIGN setting. 0=left align (default), 1=right align
text-offset	Offset the title text in pixels from the left margin (or the right if you have text-align or font-rtl set).
row-text	Provide a title text in the left column for each logical row in a multi-row item. Alternative to <i>text</i> option. You may include return values to display data. The text-align and text-offset options can be used here. See discussion here .
id	A unique alpha-numeric identifier. This is used by the DTI remote monitoring option to identify a specific item in the items section.

The visibility of each item is controlled by the *active*, *hidden* and *hide-no-result* options.

Background-interval: Set this when you want an item to continue collecting even when it is not visible. This is useful to keep charts ticking over and csv data files writing. When the page is changed and the item becomes visible, the item data is collected at the foreground *interval* rate, when the page is changed and the item becomes invisible, the item data is collected at the *background-interval* rate.

Hidden: Set the *hidden* option if you want the item to be active (collecting data) but not visible.

Hide-no-result: Set the *hide-no-result* option if you want to hide the item when there are no results. For example, hide the wifi item if wifi is not active, hide the ethernet item if ethernet is not connected. The item remains active (collecting data) so if it gets a result (wifi is connected, ethernet is connected) it will become visible and show the results.

Font-size: If you only set the default *font-size* option in the [options] or [screen-saver] section then all items will take on that font size. However, each item can individually override that *font-size* value. This override value can be an absolute value (a regular font size) or a relative value in percent. This makes it easy to scale the whole display up or down, especially for the screen saver.

Say the default font size is 10. An item may specify “*font-size:120%*” which will evaluate to 12 point. An item may specify “*font-size:80%*” which will evaluate to 8 point. If the default font size is changed to, say, 16 then the item overrides will automatically scale up as well.

Set: Use this option to assign item data to user variables.

The general form is:

```
set:myvar [myvar2=val myvar3=val ...]
```

There are two ways of using this option: the first is the traditional way of merely specifying a variable name: *set:myvar*. This assigns the display output to the variable. The second uses the equals sign to specify the raw data to be assigned to the variable: *set:myvar=%2*. This assigns a return value to the variable. This return value may be specified as a percent style return value, a WMI style %name% return value, a user variable or expression.

You can assign values to many variables within a single item and can be a mixture of the two ways as shown above. Do not use spaces within an expression, this gets the parser confused.

There is the question of what raw data is assigned from a multi-row item. The answer is this will always be the first row.

Example:

```
datetime=set:mydatetime mydate=%9 mytime=%10
```

This assigns the display output to the *mydatetime* variable, the raw date to the *mydate* variable and the raw time to the *mytime* variable.

Display: Each item has a list of available values you can display. To display a value, put it's replaceable parameter in the display template option along with any other text you want to include in the display. Refer to the [Item Reference](#) to see what replaceable parameters are available.

Example: the CPU item returns two values: cpu and kernel. To display these use %1 for the cpu value and %2 for the kernel value.

```
display:Tot: %1 Krnl: %2
```

Will display something like:

```
Tot: 5 Krnl: 2
```

If you want to use a comma in a text or display property, you must precede it with a backslash.

Incorrect Use Of A Comma

```
display:Receive:%1 bps, Transmit:%2 bps
```

Correct Use Of A Comma

```
display:Receive:%1 bps\, Transmit:%2 bps
```

Each WMI item will have as many replaceable parameters as there are properties in the result set. The replaceable parameter is the name of the property enclosed in percent signs.

Item Colors

Desktop Info has several layers of colors to give you breadth of control. At the first level is the default colors.

If you specify nothing at all, the items will be white.

To change the colors for all items, set the key word COLOR. All following items will now assume this color. To change the color for the left column only for all items, set the key word TEXT-COLOR. All following items will now assume this color in the left column. The right column will continue to be set by the key word COLOR.

To override these default colors for individual items, use the *color* and *text-color* options in the items. These item options only set the colors for that item. The default colors continue to apply to other items.

Each time you specify a new default COLOR, the default TEXT-COLOR is set to the same value. If you want a separate left column color you should specify TEXT-COLOR as well.

Likewise at the item level, setting the item *color* option will set the color for both columns. If you want a separate left column color you should specify *text-color* as well.

In other words, if you do nothing your current configuration will continue to work as it always has.

Example 1: No set colors (assumes white):

```
DATETIME=interval:10
UPTIME=interval:60
HOST=
```

Example 2: All items green:

```
COLOR=%green%
DATETIME=interval:10
UPTIME=interval:60
HOST=
```

Example 3: All items green except third item which is blue:

```
COLOR=%green%
DATETIME=interval:10
UPTIME=interval:60
HOST=color:%blue%
```

Example 4: All items will have the right column green and the left column cyan, except third item which is blue in both columns:

```
COLOR=%green%
TEXT-COLOR=%cyan%
DATETIME=interval:10
UPTIME=interval:60
HOST=color:%blue%
```

Example 5: All items will have the right column green and the left column cyan, except second item which is right column green, left column yellow and the third item which is blue in both columns:

```
COLOR=%green%
TEXT-COLOR=%cyan%
DATETIME=interval:10
UPTIME=interval:60,text-color:%yellow%
HOST=color:%blue%
```

Thresholds

Any item that displays a numerical value may be configured to change color when that item reaches or exceeds a given value. It could be an absolute value such as PAGEFAULTS or DISKQUEUE, it could be a percent such as CPU or PHYSICALRAM or a rate such as NETPACKETSRATE or DISKIO. You can define up to nine thresholds on each item.

The general form of the threshold option is:

```
threshold#:num tvalue color
```

Where:

threshold#	one of 'threshold1', 'threshold2' etc up to 'threshold9'
num	any of the returned values for the item it can be defined using an integer or percent style return value number or WMI style return value name
tvalue	threshold value that will cause the color to change it can be defined using an integer value, a percent style returned value number, a WMI style return value name, expression or user variable
color	the new color of the item it can be defined using BGR, #RGB style color values or user variable

Example: The CPU item has two return values: the cpu percent and the kernel percent. To define a threshold for the first value:

```
threshold1:1 90 0000ff
```

This means use the first return value, trigger at 90 (percent), the new color is red. I could have assigned this to threshold2, threshold3 etc., it doesn't matter.

I could define a second threshold to handle the kernel percent:

```
threshold2:2 50 %orange%
```

This uses the second return value, trigger at 50 (percent), the new color is orange.

Put both of these on the CPU item:

```
threshold1:1 90 0000ff,threshold2:2 50 0066ff
```

For WMI queries you can use the WMI property name to define the property you want in addition to the return value number.

To define a threshold which triggers when the raw value drops below a value, specify a negative threshold value. For example, the BATTERY item would have a threshold like this:

```
threshold1:%EstimatedChargeRemaining% -20 #ff0000
```

Where the threshold triggers when the battery charge falls to 20% or less.

You can use return values, user variables and expressions in thresholds. For example:

```
threshold1: 1 {{%CurrentBandwidth%/8*95/100}} %red%
```

This uses a WMI return property name inside an expression to arrive at a threshold value. In this case the bandwidth (in bits per second) is divided by 8 (bytes per second) and then converted to a percent so that the threshold is always 95% of the bandwidth regardless of what the bandwidth value currently is. The threshold color uses a user variable defined elsewhere.

Charts

This discussion refers to the new charts implemented in version 2.9. For a discussion of the old charts implemented prior to version 2.9, see the [Legacy Charts](#) section.

Any numerical value can be charted. You can use either the bar chart or line chart with a linear or logarithmic scale. The two charts are interchangeable and can easily be switched from one to the other and back again. The value to be charted can be specified a number of ways such as return value number, name or expression.

A bar chart can display one or two series. A line chart can display up to three series. The chart height can be set to any number of display rows. The chart width can be set to any number of pixels. The chart can be placed anywhere in the display template using the `%chart%` placeholder.

When you define a chart in an item, the chart is displayed for every result row. For example, if your system has three logical drives, the LOGICAL_DRIVES item will show three result rows and each result row will have a chart.

To keep track of line charts associated with multi-row results, you must specify the *row-id* option. This states the column or expression that will be used to recognise rows in the result set over time. Think of it as the primary key of the table. Each time an item is executed and rows are returned, the contents of the *row-id* column are used to match up each row with it's line chart and update it's data. For example, if you want to chart the throughput of logical drives, you would define a line chart and specify a *row-id* with the value of "`%name%`", where the `%name%` column returns values such as 'C', 'D', 'F' etc. The first time the item is run, three line charts are created, each assigned a name from the `%name%` value of 'C', 'D' and 'F'. The next time the item is run, the values in the `%name%` column, 'C', 'D' and 'F' are used to find the chart of the same name so it can be updated with the correct data.

The *row-id* option may be omitted for single row items but multi-row items must define the *row-id*. If a multi-row item does not have the *row-id* defined, the line chart will not display.

An expression may be used to construct a unique *row-id* value.

To define a chart in an item, add the "chart" key word and the type. Follow this with a series of *option:value* pairs separated by a space for the various options.

```
chart:type option:value option:value option:value ...
```

The *type* value can be either *bar2* or *line2*.

Following the *type* value, the options may be specified in any order. Most options may be omitted if not needed. You can use any of the three *seriesx* options but you must specify at least one. Only *series1* and *series2* are valid for bar charts.

Options:

scale	<i>linear</i> or <i>log</i> (required)
max	maximum value of the chart (required)
series1	the return value to chart in the first bar or line series (optional)
series2	the return value to chart in the second bar or line series (optional)
series3	the return value to chart in the third line series (optional)
color1	the color of the first line series or the low value bar color (optional, default=green)
color2	the color of the second line series or the medium value bar color (optional, default=orange)
color3	the color of the third line series or the high value bar color (optional, default=red)
threshold	the threshold value to transition from medium color to high color of the bar chart (optional, default=none)
width	the width of the chart in pixels (optional, default=auto size)
height	the height of the chart in display rows (optional, default=1)
freq	the number of pixels between data points on a line chart this has the effect of stretching out the line chart (default=1)
row-id	the return value used to identify a row for line charts (required for multi-row line charts)

Defining Chart Values

The max, series1, series2, series3 and row-id values can be defined any of the following ways:

- traditional return value number (series1:5)
- “%” style return value number (series2:%3)
- a WMI return value name (series3:%name%)
- a user variable (series1:%myvar%)
- an expression enclosed in double braces (series1:{ {%2*20/100} })

An expression can use any combination of “%1” style return values, wmi return values and user variables as well as arithmetic expressions described in the [Expressions](#) section.

The color1, color2 and color3 values can be specified as BGR, #RGB or user variables.

Examples

The Simplest Example:

A simple line chart for the CPU item:

```
chart:line2 scale:linear max:100 series1:1
```

Bar Chart Example:

To change it to a bar chart, just change the type:

```
chart:bar2 scale:linear max:100 series1:1
```

Expression Example:

A simple line chart which charts an expression:

```
chart:line2 scale:linear max:1000 series1:{{%1*10}}
```

Series Examples:

The series can be stated as a simple return value number or name:

```
chart:line2 scale:linear max:100 series1:1 series2:%12 series3:%name%
```

Negative Bar Chart

A negative bar chart is one that starts green when it is full and slowly changes to red as it empties. Think of a battery gauge. You create a negative bar chart by specifying a negative threshold value.

The big difference between a regular bar chart and a negative bar chart is with a regular bar chart you will see the color gradient appear as it goes from empty to full. So on the left you see the green, in the middle you see the orange, on the right you see the red. With a negative bar chart there is no color gradient, the entire bar starts off green when it is full, as it begins to empty and approaches the threshold the entire bar will turn orange, as it passes the threshold and on down the entire bar will turn red.

Why is it done this way? If a negative bar used a normal gradient you would always see red on the left of the bar even when the bar is full. If you were to briefly glance at DTI, you would see the red and get the impression that something needs addressing which is not so. So the bar will only show red when there is something that needs addressing (eg the battery is nearly empty).

Remember, the threshold value you set is the value at which the color will begin to change. So if you want the battery chart to be red at 20% you should set the threshold value to around -40.

Example:

```
chart:bar2 scale:linear max:100 series1:%EstimatedChargeRemaining%  
threshold:-40
```

Moving To The New Charts

For the time being, your old charts will continue to work but you should change over soon so you don't get caught out by a release that breaks everything. In 25 words or less, this is how to move your legacy charts to the new charts.

Bar Charts

The old bar chart config looks like this:

```
chart:bar linear 100 1 80, bar-colors:ff0000 0000ff 00ff00
```

Change the 'bar' to 'bar2', add the option name in front of each option and merge the *bar-colors* section to the chart section.

```
chart:bar2 scale:linear max:100 series1:1 threshold:80 color1:ff0000  
color2:0000ff color3:00ff00
```

Line Charts

The one change you'll immediately notice when moving from the old line chart to the new line chart is the default chart height is now one display row instead of two.

The old line chart looks like this:

```
chart:line linear 100 2 00ff00 4 00ffff
```

Change the 'line' to 'line2', add the option name in front of each option and add the *row-id* option.

```
chart:line2 scale:linear max:100 series1:2 color1:00ff00 series2:4  
color2:00ffff row-id:1
```

Make sure the *row-id* option points to a return value with unique data such as index number, drive name etc. It can be a number for standard DTI items or a return value name such as *%name%* for WMI items. It can also be an expression that results in a unique value. If the item will only return a single row, the *row-id* is not required.

Legacy Charts

This discussion refers to the original bar and line charts implemented prior to version 2.9. With version 2.9, a more sophisticated charting system is available. See the [Charts](#) section for details.

Any numerical value can be charted. You can use either the bar chart or line chart with a linear or logarithmic scale. The value to be charted can be specified a number of ways, see the Chart Series section below. See the Chart Positioning section below for important information regarding bar chart positioning.

The general form for the bar chart option is:

```
chart:bar scale max series [[-]tvalue]
```

Where:

bar	displays a bar chart
scale	'linear' = linear scale 'log' = logarithmic scale
max	maximum value of the chart
series	the value to display
tvalue	threshold value of the bar chart (optional)

If the threshold, tvalue, is a positive value, the bar will display a gradient as the value increases towards and past that value, a 'disk used' gauge for example. If the threshold is a negative value, the whole bar will change color when the value decreases to that value or below, a battery gauge for example.

See also, the [Bar Chart Colors](#) section below.

The general form for the line chart is:

```
chart:line scale max series1 color1 [series2 color2]
```

Where:

line	displays the line chart
scale	'linear' = linear scale 'log' = logarithmic scale
max	maximum value of the chart
series1	the value to display
color1	color of series1
series2	the value to display (optional)
color2	color of series2

Chart Positioning

The chart can be placed anywhere within the item display by placing the *%chart%* marker in the *display* template option. For example:

```
... chart:bar linear 100 7 80,display:%7|%chart%|%1 %2 %3
```

This will place it on the second display line for that item. In this case, it makes sense to place the chart which is charting the return value %7 right after the text displaying that value. This works well for LOGICALDRIVES.

It can also be placed on a display line following text:

```
... chart:bar linear 100 1 75,display:%1 %chart%
```

This works well for CPUUSAGE for example by placing the bar chart immediately after the value.

You may place addition text on the same display line after the chart and the chart will shrink accordingly. If there is no text following the chart on the same display line, the bar chart will grow to take up the remainder of that display line. For example:

```
... chart:bar linear 100 1 75,display:%1 %chart% %2|%3
```

If you define a bar chart for an item but do not place the marker in the *display* template, the bar chart will be placed at the end which may look fine or it may be too small. Place the pipe symbol, '|', prior to the chart marker to force it on to a new line.

Conversely, if you place the chart marker in the display template without defining a chart, nothing will be displayed.

For now, the line chart continues to display after the item display. I expect this will follow the %chart% marker in the future.

Chart Series

The exact value charted can be specified a number of ways. Traditionally, this was the number of the item return value but can also be any of the following:

- a regular item return value number (chart: bar linear 100 7 75)

- a “%” style item return value number (chart: bar linear 100 %7 75)

- a wmi return value (chart: bar linear 100 %scale% -25)

- a user variable (chart: bar linear 100 %myvar% -25)

- an expression enclosed in double braces

- (chart: bar linear 100 {{%freespace% * 100 / %size%}} -25)

An expression can use "%1" style return values, wmi return values and user variables.

Example using wmi return values in an expression:

```
WMI=TEXT:Drive C: Free %,style:b,namespace:root\  
cimv2,query:Win32_Logicaldisk where deviceid = "C:", \  
chart:bar linear 100 {{%freespace% * 100 / %size%}} -25,display:  
{{round(%freespace% * 100 / %size%)}}
```

Max Value

The max value can be specified a number of ways. Traditionally, this was a regular absolute value but can also be any of the following:

- a “%” style item return value number (chart: bar linear %8 %7)

- a wmi return value (chart: bar linear %total% %value%)

a user variable (chart: bar linear %mymaxvar% %7)

an expression enclosed in double braces

(chart: bar linear { {%freespace%+%usedspace%} } %freespace%)

An expression can use “%1” style return values, wmi return values and user variables.

Remember that every time the max value changes the chart is reset. For a line chart this means the current data history is lost and the chart starts again.

Negative Bar Chart

A negative bar chart is one that starts green when it is full and slowly changes to red as it empties. Think of a battery gauge. You create a negative bar chart by specifying a negative threshold value.

The big difference between a regular bar chart and a negative bar chart is with a regular bar chart you will see the color gradient appear as it goes from empty to full. So on the left you see the green, in the middle you see the orange, on the right you see the red. With a negative bar chart there is no color gradient, the entire bar starts off green when it is full, as it begins to empty and approaches the threshold the entire bar will turn orange, as it passes the threshold and on down the entire bar will turn red.

If a negative bar used a normal gradient you would always see red on the left even when the bar is full. This creates the impression that something needs addressing which is not so. So the bar will only show red when there is something that needs addressing (eg the battery is nearly empty).

Remember, the threshold value you set is the value at which the color will begin to change. So if you want the battery chart to be red at 20% you should set the threshold value to around -40.

Chart Examples

Example: A simple battery gauge:

```
chart:bar linear 100 1
```

A bar chart with a linear scale, a maximum value of 100 (percent) and uses the first return value which is percent charge remaining.

You can have the bar begin changing to red when it drops to a certain critical value:

```
chart:bar linear 100 1 -40
```

The negative threshold tells the bar to begin changing color when the value drops below 40%.

Example: Cpu chart:

```
chart:line linear 100 1 00ff00 2 0000ff
```

A line chart with a linear scale, a maximum value of 100, the first series uses the first return value (cpu percent) and has a color of green, the second series uses the second return value (kernel percent) and has a color of red.

Logarithmic scales apply a log base 2 calculation to the value before charting it. This transforms a value of 1000 to about 10, a value of 10000 to about 13. This allows you to have sensible charts ranging from zero to huge numbers such as disk transfer speeds or network speeds. For a network speed chart with typical speeds of 10MiBps you would have a log chart with a maximum value of around 26. For a throughput of 1GiBps you would have around 30. For more information, Google is your friend.

Line charts always consume two rows below the item result text but bar charts are normally appended to the end of the result text display. If the display text is long the bar chart can be tiny. You can shift the bar chart on to the next row by adding a pipe symbol to the end of the display template:

```
PHYSICALRAM=interval:5,color:ffd0e0,threshold1:3 80 0000FF,chart:bar  
linear 100 3 80,display:%1[3.1b]B / %2[3.1b]B (%3% used)|%chart%
```

Legacy Bar Chart Colors

You can define custom colors for the legacy bar charts. There are three colors:

chart area	default color
below threshold	green
threshold	orange
above threshold	red

Use the *bar-colors* common option to define the three colors in either 6 character bgr form or 7 character rgb form separated by spaces.

For example, the CPU item might have a bar chart defined:

```
CPU=chart:bar linear 100 1 80
```

Add the bar-colors option thus:

```
CPU=chart:bar linear 100 1 80,bar-colors:ff0000 0000ff 00ff00
```

This will set the three colors to blue, red and green respectively.

You can also use user variables set earlier in the ini file.

```
bar-colors:%Blue% %Orange% %Magenta%
```

Data Logging

You can log individual items by adding 'csv:filename'. For example:

```
csv:c:\cpu.csv
```

This writes the data to a csv formatted file where the cells are enclosed in quotes and separated by commas and the first row contains column identifiers.

By default, the data is written in it's raw form. You can write the formatted data by adding 'csvdatatype:1'.

This example writes raw data to a file in the root directory:

```
PAGEFAULTS=interval:10,csv:\pagefaults.csv
```

This example writes formatted data:

```
PAGEFAULTS=interval:10,csv:\pagefaults.csv,csvdatatype:1
```

User Variables

User variables are a way to store data and text so it can be used later. The user variable names are not case sensitive. There are three ways to set a user variable:

1. Using the "set" option in any item. For example:

```
HOST=set:hostname
```

stores the display of the HOST item in a variable called "hostname". The variable will be available only after the item has been evaluated.

2. You can store raw item values using the “set” option with the equals sign. For example:

```
DATETIME=set:mydate=%9 mytime=%10
```

stores the raw date value into a variable called “mydate” and the raw time value into a variable called “mytime”.

3. Using the SET item you can store anything you can type in. For example:

```
SET=key:MyTestKey1,value:MyTestValue1
```

stores the value of "MyTestValue1" into a variable called "MyTestKey1".

Having stored these values into user variables, you can display them using any item's display by typing the variable name enclosed in percent signs. For example:

```
TEXT=text:Host/User,display:%hostname%/%username%
```

displays the "hostname" and "username" variables together on a single line.

```
Host/User          DESKTOP-PC/Glenn
```

This is the complete example.

```
HOST=set:hostname,hidden:1
USER=set:username,hidden:1
TEXT=text:Host/User,display:%hostname%/%username%
```

Here's the same thing done a slightly different way. It does the USER item first and stores that result then does the HOST item and displays both at once.

```
USER=set:username,hidden:1
HOST=text:Host/User,display:%1/%username%
```

Here's another example. It combines boot time and up time into a single line display. Notice both the UPTIME interval and the TEXT interval are set so it continues to update.

```
UPTIME=set:uptime,display:%1d :%2h :%3m :%4s
BOOTTIME=display:%9 (Up: %uptime%)
```

In this example we want to read a small text file containing important information that gets updated by the domain admin. First we monitor the file for changes, read the first line into a user variable then display that variable. In this case it's the support extension number that changes daily. It could be asset or other information the domain support people want to push out to computers.

```
FILE=interval:60,type:text,text:Contact,file:\contact.txt,display:
%2,set:supporttext,hidden:1
TEXT=interval:60,text:Support,display:Call Support on ext %supportext%
```

An interesting thing to note is the TEXT item itself can have the set parameter. This means you can cascade user variables or build them from other variables. If you can find a use for this I'm listening.

The SET item can also include user variables. For example:

```
SET=key:var1,value:complete
SET=key:var2,value:output
SET=key:var3,value:%var1% %var2%
TEXT=display:%var3%
```

This will display something like:

```
complete output
```

You can also use variables to predefine colors to be used throughout the configuration. At the top of the [items] section, place a series of SET items defining color names and values. You can then use these color variables instead of direct values in the remaining configuration items. See the included ini files for examples and the [SET](#) item.

```
SET=key:white,value:#ffffff
SET=key:cyan,value:#00ffff
SET=key:green,value:#00ff00
COMMENT=color:%white%,text:Desktop Info
```

Note, a variable may reference other variables but don't let a variable reference itself. This will cause a stack overflow and crash the program.

Conditional Visibility

You can control an item's visibility using User Variables and the IF item. You use the IF item to compare two values, if the comparison returns true the following item is displayed, if the comparison returns false the following item is not displayed. The two values can be user variables, numbers or literal character strings.

For example, if the BATTERY item returns no result (ie there's no battery), then you don't need to display it. To achieve this, you run the BATTERY item using the 'hidden' option and the 'set' option. Then you use an IF item to compare that result with the "noresults" text string. On the next line you use a TEXT item to display the results of the BATTERY item. If the IF comparison returned true the TEXT item is displayed, otherwise it is skipped.

```
BATTERY=interval:60,set:battery,display:%1%,hidden:1
IF=value1:%battery%,value2:<n/a>,comparator:ne
TEXT=interval:60,text:Battery,display:%battery%
```

See the [IF](#) item for available options.

You can also dynamically control visibility of an item using the [hide-no-result](#) option.

Left Column Text

Each item has a default text that will be displayed in the left column. You can override this text in each item with the *text* option. For example, the DATETIME item has a default text of "Date/Time". However, you may choose to display only the date so you might add the *text* option to change the title text.

```
DATETIME=interval:10,text:Date
```

You can also use the *text* option to display text in your own language.

The default text and the *text* option will only display once for an item regardless of how many logical rows the item has. For example, the LOGICALDRIVES item will display the default "Logical Drives" text once on the very first display line of the item. The left column remains blank for all succeeding display lines and all logical drives that are displayed.

To display a title text on the first display row of each logical row, use the *row-text* option. You can include return values to display row data in the same way as the display template.

```
LOGICALDRIVES=interval:5,row-text:Drive %1:,display:%7[1.1f]%
```

This will result in a title text in the left column for each drive in the results.

```
Drive C:          27%
Drive D:          30%
```

The *text-align* and *text-offset* options may be used for the default text and also the *text* and *row-text* options to position the text within the left column.

Remote Monitoring

To monitor a remote instance of Desktop Info, the first thing to do is switch on the [internal http server](#) of Desktop Info on the machine you wish to monitor. To do this there are just two options you need to set in the ini file: *server-ip* and *server-port*.

```
[options]
server-ip=xxx.xxx.xxx.xxx
server-port=xxx
```

The *server-ip* is the ip address of the machine. Usually this is something like 192.168.0.12 or 10.0.0.15 or something along those lines. This is required as any given machine may have any number of network interfaces. The *server-port* can be any port not in use. For example port 80 is a common http port. If the http server fails to start then either the ip address is not valid or the port is in use. You might try different ports, I'm currently using 8192. Setting these two options is all you need to do to get the http server running. I haven't yet tested this across network boundaries, your mileage may vary.

Next, you need to give each item you wish to retrieve from the server a unique *id* string. Only those items with this unique *id* set are available for remote retrieval. For example, if you want to retrieve the raw data for the CPU item, set the *id* option, in this case I've given the item an id of *cpu1*.

```
CPU=interval:1,id:cpu1,display:Tot: %1[2.0f]%\, Krnl: %2[2.0f]%
```

If you want to monitor items that are not on the active page, you will also need to set the *background-interval* option.

The third thing you need to do is add an entry to the [items](#) of the client instance of Desktop Info (the instance that will remotely retrieve the raw data) for each item you wish to retrieve. In the item options, specify the unique *id* that identifies the item you want to retrieve that you added above.

```
DTI=host:10.0.0.15,port:8080,timeout:500,id:cpu1,interval:10,display:
%1 (pid:%2) %3%
```


Desktop Info - Manual

This item connects to Desktop Info on the host 10.0.0.15 and retrieves raw data for the item with the id of *cpu1*. You are now free to use the display template to format the raw data as desired, add thresholds and maybe a chart.

Keep in mind that WMI property names are not brought over so you will need to refer to them by position number (%1, %2 etc) rather than name in the display template and other options.

Item Reference

The following is a complete reference of available items and their options and return values. See also the list of [common options](#).

Return values are listed as replaceable parameters; use these in the display template property to display these values. See the formatting section earlier for more information and the sample ini files for examples. Any item that returns a numerical value can be charted and have thresholds. Use the item return value numbers for the chart and threshold series.

ALLIPADDRESS

Shows all IP addresses assigned to all interfaces.

Available Options:

activeonly	0=all interfaces (default) 1=only active interfaces
filter	include or exclude interfaces based on description
multirow	0=comma delimited on a single line 1=one ip address per line (default)

Return Values:

%1	ip address
%2	subnet mask
%3	prefix length

Example:

```
ALLIPADDRESS=interval:30,activeonly:1
```

The filter option may contain *includes* denoted by a '+' sign and *excludes* denoted by a '-' sign.

Example:

```
filter:+wireless-virtual
```

Shows all interfaces that contain the keyword "wireless" in the description and will exclude interfaces that contain the keyword "virtual" in the description.

If there is more than one filter match, they are evaluated left to right and the last one wins. For example, "filter:+wireless-wire" will include all interfaces with "wireless" and then exclude any interface with "wire", resulting in no wireless interfaces.

An interface must match all the given filters in order to be displayed.

BOOTTIME

Shows the time of day the host was booted.

Return Values:

%1	day of week
%2	date
%3	month
%4	year
%5	hours
%6	minutes
%7	seconds
%8	unix time
%9	short date
%10	short time
%11	long date
%12	long time

Unix time is defined as the number of seconds that have elapsed since 00:00:00 Thursday 1st January 1970 UTC minus leap seconds.

The short and long date and time return values are formatted according to the local Windows regional settings. The default display template is:

```
display:%9 %10
```

CMD

Run a shell command and display the output and/or exit values. The command can be a Cmd.exe shell command or batch file, a PowerShell.exe command or script or pretty much any tool or application. Desktop Info reads the STDOUT output directly and stores it in return value %4 as a single blob of text. If the command does not write to STDOUT then %4 will be empty and the only clue to success or failure will be return values %1, %2 and %3.

If %4 is a single numeric value then it can be used in a chart.

Available Options:

file	the executable to run
	eg. cmd.exe, powershell.exe, myapp.exe etc

directory	the current working directory for the executable Important: if the directory ends in a backslash, leave a space before the comma so it doesn't seem to be an escaped comma
parameters	the command line arguments to pass to the executable
show-window	show the command window (possibly useful diagnostic) 0=hide the window (default) 1=show the window
wide	0=display in right column (default) 1=display over both columns
code-page	assume the returned text is Unicode and decode it using the given code page
trim	try to trim leading and trailing white space and line feeds from the output (1=trim, 0=no trim, default=0)
read-as-csv	assume the cmd output is in csv format and attempt to parse it into rows of data (0=normal text,1=csv text, default=0). See below.

Return Values (if read-as-csv=0):

%1	CreateProcess result - whether DTI succeeded in creating the shell process regardless of the outcome of the command itself 0 = failed 1 = succeeded
%2	CreateProcess error if %1 is 0, this gives the error code
%3	command exit code if %1 is 1, this returns the exit code of the given command eg the command shell <i>errorlevel</i> or the PowerShell exit code or the application exit code
%4	the command STDOUT output text. This is a single blob of text and can't be parsed or interpreted in any way. If this text equates to a single numeric value then it can be used in a chart.

By default, return value %4 is displayed which is the STDOUT text created by the command. This, of course, implies that the command must write to STDOUT and only the text written to STDOUT will be captured. Carriage Return / Line Feed pairs and single line feeds are handled for display. If the command fails you can use %1, %2 and %3 to help debug it. If you don't need any kind of display, set the *hidden* option.

Some of the show-window options are:

0	hide (default)
1	show normal
2	activate and show minimized
3	activate and show maximized
4	show but don't activate

For a full list see: <https://docs.microsoft.com/en-us/windows/desktop/api/winuser/nf-winuser-showwindow>.

To display Unicode text, the target command must be Unicode aware. Cmd.exe is not Unicode by default, you must supply the /U command line argument.

Example – Internal Command with Unicode:

```
CMD=file:cmd.exe, parameters:/u /c echo äöüß
```

The /C and the command must be together at the end of the parameters option.

Desktop Info attempts to determine if the returned text is UTF-8 and handle it accordingly. If this test fails, the text is assumed to be ASCII text. In the case of Powershell, the returned text may be UTF-16. DTI doesn't test for this. You can force multi-byte decoding by supplying the code page number using the *code-page* option.

Example – Powershell Script with Decoding:

```
CMD=file:powershell.exe, parameters:-File test.ps1, code-page:437
```

Code page 65001 is said to be synonymous with UTF-8 so if the default test doesn't correctly identify it, you can try forcing a UTF-8 decode by setting the "*code-page:65001*" option.

There is a comprehensive list of known code pages at:

<https://docs.microsoft.com/en-us/windows/win32/intl/code-page-identifiers>

Example – Powershell Command:

```
CMD=file:powershell.exe,directory:d:\temp,parameters:-noprofile -  
command Get-WMIObject -Class Win32_Processor -ComputerName . | Select-  
Object -Property SystemName | Format-List | Out-String | ForEach-  
Object { $_.Trim() }
```

This executes a PowerShell command directly. The parameters option is quite extensive and consists of the command to execute followed by commands for getting the output into a state suitable for display in Desktop Info.

There is also a good Powershell example in the desktopinfo-advanced.ini file.

If your Powershell script fails to execute, it may be due to the Powershell execution policy which pretty much disables everything by default. To allow DTI to run Powershell scripts I type the following command in to a Powershell command line (you only need to do this once on any given machine):

```
Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
```

CSV Output

To display csv formatted output in CMD, for example nvidia-smi.exe, add the “*read-as-csv:1*” option.

Additional Options:

csv-header	0=there is no header row
	1=there is a header row
	default=1
csv-field-count	0=variable field count
	>0=fixed field count
	default=0

The first line of the output text is assumed to be the csv column headers and each successive row is data. DTI uses these column headers to identify return values in the same way as the WMI item, for example, *%name%*, *%cpu%* etc. If there is no header row, use the *%1* style return values to display the return data. The field separator is comma. If your external tool requires commas on the command line to separate the arguments, as nvidia-smi does below, you should create a batch file and call that from the CMD item.

The csv rows may or may not contain an equal number of values. This is not fatal, the short rows may display values as not available. You can force all rows to have the same number of values by using the *csv-field-count* option, though this is normally not necessary.

The fields may be quoted using double quotes or not or any combination.

Any return values that equate to numerical values may be used in expressions and charts.

The normal CMD return values, *%1 %2 %3 %4*, are not used. The return values are the csv contents.

An example output from nvidia-smi.exe:

```
$ nvidia-smi --query-gpu=gpu_name,gpu_bus_id,vbios_version --  
format=csv  
  
name, pci.bus_id, vbios_version  
GRID K2, 0000:87:00.0, 80.04.D4.00.07  
GRID K2, 0000:88:00.0, 80.04.D4.00.08
```

You can use `%name%`, `%pci.bus_id%` and `%vbios_version%` to identify the return values or `%1`, `%2` and `%3`.

You can call a batch file from CMD and use the batch file to construct a data set using all the tools at your disposal and echo the whole lot back to DTI.

```
CMD=interval:5,file:cmd.exe,parameters:/C test.bat,read-as-csv:1,csv-  
header:0,row-text:%1,display:%2  
  
test.bat  
@echo OS, %OS%  
@echo Processor, %PROCESSOR_IDENTIFIER%  
@echo User, "%USERNAME%"
```

COMMENT

Allows you to put any fixed text on the display. Makes a nice banner and section header. You can also use this to create a blank line to visually separate blocks; just create a COMMENT item with no text.

Available Options:

text	Text to display over both columns
------	-----------------------------------

CONTROL

This item provides a clickable control on the screen to open any given universal resource. This can be an internet resource such as a web page, a network resource such as a network shared folder or shared document or a local resource such as a folder, document, application, batch file, script. Pretty much anything you might type in to the Windows Run dialog should work.

<https://www.w3.org/wiki/URI>

Available Options:

type	0=button (default), 1=hyperlink
button-color	the background color of the button
	background color of the hyperlink during mouse hover
hover-color	the text color when the mouse hovers over the hyperlink

uri	the resource to open or command to execute
args	any command arguments
wide	0=right column (default), 1=both columns
left	left position of control (default=0)
width	fixed control width in pixels (default is auto-sized) only for buttons, hyperlinks are always auto-size
control-style	applies to hyperlink controls, see below

The common option *style* can be used to control bold, underline and italic.

When working with hyperlink controls there are two style options to consider. If neither style is specified then both the left column text and the hyperlink control are normal but the hyperlink is always underlined. If only the style option is specified then it is applied to both left column text and the hyperlink control but the hyperlink is always underlined. If control-style is specified then it is applied to the control only independently of the left column text style and it may or may not be underlined.

The text color is defined by the common *color* option. The button and hyperlink controls are identical in function, it's purely a matter of personal preference. When the user clicks on the control, either the resource is opened or nothing happens.

Any document with an associated file type (doc, txt, pdf etc) or protocol (http, ftp, mailto etc) will open in the default application configured in Windows.

User variables can be used in the *display*, *uri*, *button-color* and *hover-color* options.

Environment variables used in the uri option are resolved prior to execution.

Example:

```
CONTROL=text:,display:Documents,uri:%userprofile%
```

CONTROL2

This item is identical to the CONTROL item except that it doesn't display on it's own display row. Instead it will display on the same display row as the most recent CONTROL item. In this way we can have multiple controls on a single display row. If there is no previous CONTROL item, the CONTROL2 item cannot be displayed.

Think in terms of a control group (a group of controls) where the CONTROL item is the parent and CONTROL2 items are the children. The CONTROL item is always the first button on the row and the CONTROL2 items follow it on the same row.

When an IF item changes the state/visibility of a CONTROL, the child CONTROL2 items will follow.

CONTROL2 is not regarded as an 'item' in it's own right. Therefore you can't use it independently and you can't use an IF item with it.

To use CONTROL2, begin with a CONTROL item as you normally would. Set the *wide* option if you want the controls to span both columns. Set the *width* option if you want the control to have a fixed width.

Follow this with one or more CONTROL2 items. Set the *left* option so that they're offset from the previous button. Optionally, set the *width* option so that they have the same fixed width as the CONTROL item.

```
COMMENT=text:System Tools,style:iwb,color:%white%
CONTROL=wide:1,width:115,display:Device Manager,uri:devmgmt.msc
CONTROL2=left:125,width:115,display:Documents,uri:%userprofile%
CONTROL2=left:250,width:115,display:Control Panel,uri:control
```

In the above example, the CONTROL item does not specify the *left* option so it defaults to 0. The following CONTROL2 items set the *left* and *width* options so that the buttons are separated by 10 pixels.

To set up a nice looking grid of buttons of, say, 3 buttons across, you would start with a CONTROL followed by 2 CONTROL2 items then another CONTROL item to start the next row, followed by 2 more CONTROL2 items etc. for as many display rows as needed. The above example could be duplicated for additional rows of buttons.

See the desktopinfo-advanced.ini file an example.

CORETEMP

This item reads the shared memory area of Core Temp. This tool must be running, visible or hidden, for Desktop Info to interact with it. Desktop Info does not need to be run as administrator using this approach. Best of all it's much more likely to succeed on any computer than the WMI method.

Get Core Temp at: <http://www.alcpu.com/CoreTemp/>

Hint: Pay attention during the installation process!

Return Values:

%1	Temperature
----	-------------

CPU

Shows percentage of all cpus in the system. It will always show 0-100% regardless of how many cpus there are.

Return Values:

%1	cpu
----	-----

%2	kernel
%3	processor queue length

CPUCOUNT

Shows the number of cpus in the system. This also reflects cores per cpu.

This item requires the Processor performance counter. If this counter is not available, the item returns no results.

Return Values:

%1	cpu count
----	-----------

CPUTYPE

Shows information about the brand and type of cpu.

Return Values:

%1	cpu information
----	-----------------

CPUUSAGE

Shows the usage for each cpu or core.

This item requires the Processor performance counter. If this counter is not available, the item returns no results.

Available Options:

maxrows	maximum number of cpus/cores to display
---------	---

Return Values:

%1	cpu/core number
%2	cpu/core usage

DATADUMP

Writes a csv data file for every active item. The data is the raw or formatted data from the most recent collection of each item. All csv files are written to the one directory. The file name consists of either the item lid or id and the date and time of the data collection.

Available Options:

dir	the directory to write the csv files
datatype	0=raw, 1=formatted

Return Values:

%1	the number of active items written
%2	date/time of last data dump

DATETIME

The current date and time of the host.

Return Values:

%1	day of week
%2	date
%3	month
%4	year
%5	hours
%6	minutes
%7	seconds
%8	unix time
%9	short date
%10	short time
%11	long date
%12	long time

Unix time is defined as the number of seconds elapsed since 00:00:00 Thursday 1st January 1970 UTC minus leap seconds.

The short and long date and time return values are formatted according to the local Windows regional settings. The default display template is:

```
display:%9 %10
```

DIRECTX

Shows the Direct X version.

Return Values:

%1	Direct X version
----	------------------

DISPLAYCONTROLLER

Shows the current display controller information.

Return Values:

%1	display controller
----	--------------------

DNSSERVER

Shows all DNS servers over all network adapters.

Available Options:

multirow	0=comma delimited on a single line
	1=one ip address per line (default)

Return Values:

%1	dns servers
----	-------------

DOMAIN

Shows the current domain.

Return Values:

%1	domain name
----	-------------

DOMAINCONTROLLER

Shows the current domain controller.

Return Values:

%1	domain controller
----	-------------------

DTI

Retrieves raw data for a given item from a remote instance of Desktop Info. The remote instance must have the [internal http server](#) running. The raw data from the remote item is stored in the local DTI item as if it were collected locally. This means all common options are available such as display formatting, thresholds, row-text, charting etc.

Available Options:

host	the remote host. This can be a host name or ip address.
port	port where the http server is running
timeout	http connection timeout in milliseconds (default=2000)
id	the item id for the raw data you wish to retrieve.

See the discussion at [Remote Monitoring](#) for a detailed explanation of setting up remote monitoring.

Example:

```
DTI=host:10.0.0.15,port:8080,timeout:500,id:cpu1,interval:10,display:
%1 (pid:%2) %3%
```

ENVVAR

Shows the given environment variable. The variable must be set in the parent environment at the time the Desktop Info process was started.

Available Options:

key	the environment variable to return
-----	------------------------------------

Return Values:

%1	the key of the environment variable
----	-------------------------------------

%2	the value of the environment variable
----	---------------------------------------

The default *text* value is “%1” and the default *display* template is “%2” so the entry can be as simple as:

```
envvar=key:SystemDrive
```

EVENTLOG

Returns the most recent entry for the given event log.

Available Options:

log	name of the event log to retrieve (application, system, security etc)
-----	--

Return Values:

%1	date/time generated
----	---------------------

%2	event id
----	----------

%3	event type
----	------------

%4	task category
----	---------------

%5	source name
----	-------------

%6	computer name
----	---------------

%7	event data
----	------------

The following example will monitor the system event log:

```
EVENTLOG=interval:10,log:System,display:%1|%5|%7
```

FILE

Monitors files and folders for changes to size, write time or version number.

User variables and environment variables may be used in the file name option. User variables may be used within the text of the file for the text monitor type.

Available Options:

type	monitor type (text, size, time, version)
file	file or folder name

Return Values:

%1	the file name being monitored
%2	text: the first line of text in the file
	size: the file size
	time: the last write and access times
	version: version number of the executable

This monitor will work for local and network files and folders. When monitoring a directory make sure there is no trailing backslash.

This example displays the first line of the given file whenever it changes:

```
FILE=interval:10,type:text,file:c:\temp\semaphore.txt,display:%2
```

This example shows the executable version number whenever it changes:

```
FILE=interval:30,type:version,file:C:\desktopinfo\
desktopinfo.exe,display:%2
```

FILE2TEXT

Displays the contents of the given text file containing 'key=value' pairs. The file may contain any number of lines of text. Any line that contains an equals '=' sign will be displayed; the text before the equals sign is displayed in the left column, the text after the equals sign is displayed in the right column. You can display a single entry from the file by using the 'key' option. Set the interval if you want to update it at regular intervals. The display template is not used.

User variables may be used in the file name option and key option and also within the text of the file. If you don't want to display the key in the left column, set the *hide-key* option.

Supports DOS (CRLF) and Unix (LF) text files and multi-byte Unicode text (UTF).

Available Options:

file	name of file containing <i>key=value</i> pairs
------	--

key	display only the value given by key
hide-key	don't display the key in the left column

Example:

```
FILE2TEXT=interval:3600,file:\assetinfo.txt
```

FILECONTENTS

Displays the contents of the given text file. Set the interval if you want to update it at regular intervals. The display template is not used.

User variables may be used in the file name option and also within the text of the file.

Supports DOS (CRLF) and Unix (LF) text files and multibyte text (UTF).

Available Options:

file	name of file to display
wide	0=display in right column (default) 1=display over both columns

Example:

```
FILECONTENTS=interval:3600,text:Instructions,file:instructions.txt
```

FILECSV

Reads the given csv file and displays it as multi-row data.

Available Options:

file	the external csv file to read
csv-header	whether the first row of the file is the csv header row 0=first row is not a header row 1=first row is a header row default=1
csv-field-count	the number of csv fields in the file 0=variable field count (we don't care) >0=fixed field count default=0

The first row of the file text is assumed to be the csv column headers and each successive row is data. DTI uses these column headers to identify return values in the same way as the WMI item, for

Desktop Info - Manual

example, `%name%`, `%cpu%` etc. If there is no header row, use the `%1` style return values to display the return data. The field separator is comma.

The csv rows may or may not contain an equal number of values. This is not fatal, the short rows may display values as not available. You can force all rows to have the same number of values by using the `csv-field-count` option, though this is normally not necessary.

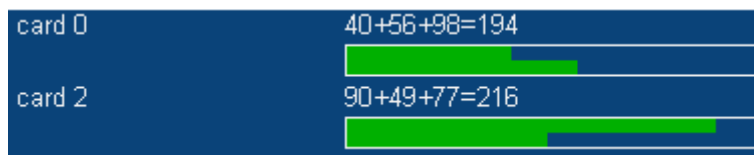
The fields may be quoted using double quotes or not or any combination.

Any return values that equate to numerical values may be used in expressions and charts.

An example csv display with expressions and charts:

```
"name", gpu, temp, speed
card 0, 40, 56, 98
card 2, 90, 49, "77"
```

```
FILECSV=interval:5,file:test.csv,row-text:%1,display:%2+%3+%4={{%2+%3+%4}},chart:bar2 scale:linear max:100 series1:%2 series2:%3
```



FILEEXIST

Checks if the given file exists and returns true or false. Use the [boolean formatting](#) to make a nice display.

User variables may be used in the file name option.

Available Options:

file	file name to check
------	--------------------

Return Values:

%1	True or False
----	---------------

Example:

```
FILEEXIST=interval:10,file:\glennisawesome.txt,text:Is Glenn
Awesome?,display:%1[b:Absolutely!:Not!]
```

This tests for the file "`\glennisawesome.txt`". If it exists the display will be:

"Is Glenn Awesome?"	Absolutely!"
---------------------	--------------

If the file doesn't exist, the display will be:

"Is Glenn Awesome?"

Not!"

HTTPGET

Makes a HTTP GET request to the given source url and displays whatever it returns. You can give it any url but it's really only useful for returning a short plain text message such as getting your public ip or if you have a http server that provides information. There is no default, if you don't specify a url, it will return an error.

Available Options:

source

url to send request

Return Values:

%1

returned value from http request

Example:

```
HTTPGET=interval:600,source:http://plain-text-ip.com/
```

Here's a few other sites that return your public ip:

<https://ifconfig.co/ip>

<http://ipecho.net/plain>

<https://api.ipify.org/>

<https://wtfismyip.com/text>

<http://ident.me/>

<https://myexternalip.com/raw>

<http://plain-text-ip.com/>

HOST

Host name of the local machine. This can provide a variety of names from local NetBIOS host name through to fully qualified domain names. The first returned value is derived from GetComputerName and the remaining are derived from GetComputerNameEx.

Return Values:

%1

the NetBIOS name of the local computer

%2

dns domain name

%3

fully qualified dns domain and host name

%4

dns host name

%5

NetBIOS name

%6	physical dns domain name
%7	physical fully qualified domain and host name
%8	physical dns host name
%9	physical NetBIOS name

See the following for more information:

<https://docs.microsoft.com/en-us/windows/win32/api/winbase/nf-winbase-getcomputernamew>

<https://docs.microsoft.com/en-us/windows/win32/api/sysinfoapi/nf-sysinfoapi-getcomputernameexw>

IEVERSION

Shows the Internet Explorer version.

Return Values:

%1	ie version
----	------------

IF

Compares two values and if true, the following item is displayed, if false, the following item is not displayed. If both values are valid numerical values, they are compared as numbers, if not they are compared as character strings. See [Conditional Visibility](#).

IF cannot be used on CONTROL2 items as they are part of a group. Use it on the parent CONTROL item.

Available Options:

value1	the first value to compare	
value2	the second value to compare	
comparator	how to compare	
	eq	value1 and value2 are equal
	ne	value1 and value2 are not equal
	gt	value1 is greater than value2
	lt	value1 is less than value2
	ge	value1 is greater than or equal to value2

le	value1 is less than or equal to value2
contains	value1 contains value2
notcontains	value1 does not contain value2

To perform a case sensitive comparison, prefix any of the comparators with 'c'. For example, 'eq' is case *insensitive* 'equals' and 'ceq' is case *sensitive* 'equals'.

You can include variables and expressions in the values so you can do some math on the fly.

```
IF=value1:{{%var1%+%var2%}},value2:30,comparator:eq
```

The following example shows how to use IF to choose between two buttons depending on the version of Windows.

```
# get the OS build
OSBUILD=set:OSBuildVer,display:%1,hidden:1

# if it's Win 10 1903/1909
IF=value1:%OSBuildVer%,value2:6.3.18,comparator:contains
CONTROL=text:Support:,display:Helpline,uri:https://intranet/help-win19

# if it's Win 10 1803/1809
IF=value1:%OSBuildVer%,value2:6.3.17,comparator:contains
control=text:Support:,display:Helpline,uri:https://intranet/help-win18
```

LOADTIME

The time of day Desktop Info was loaded.

Return Values:

%1	day of week
%2	date
%3	month
%4	year
%5	hours
%6	minutes
%7	seconds
%8	unix time
%9	short date
%10	short time
%11	long date
%12	long time

Unix time is defined as the number of seconds that have elapsed since 00:00:00 Thursday 1st January 1970 UTC minus leap seconds.

The short and long date and time return values are formatted according to the local Windows regional settings. The default display template is:

```
display:%9 %10
```

LOGICALDRIVES

Displays storage and performance metrics for selected logical drives. Shows all fixed and mounted logical drives. It does not show optical, mapped or network drives.

Available Options:

include	drives to include drive letters only, do not include colons (eg. "cdef") blank=include all drives (default) if specified, only these drives are displayed
exclude	drives to exclude drive letters only, do not include colons (eg. "abefg") blank= do not exclude any drives default is to exclude floppy drives ("ab") if specified, only drives that pass the include filter can be excluded
drive-size	include only drives that are at least this big in bytes default is 0 (include drives of any size)
diskio	retrieve disk i/o performance data 0=no 1=yes (default)
max	maximum number of drives to display (default is all drives)

Return Values:

%1	the assigned drive letter
%2	file system
%3	volume label
%4	serial number

%5	total capacity in bytes
%6	used size in bytes
%7	used size in percent
%8	free size in bytes
%9	free size in percent
%10	size available to user in bytes
Performance Data	
%11	read bytes per second
%12	read operations per second
%13	write bytes per second
%14	write operations per second
%15	average seconds per read operation
%16	average seconds per write operation
%17	queue length
%18	percent disk time
	(goes greater than 100%, I don't know why, perfmon does too)
%19	percent idle time

Useful Performance Boundaries

Average Seconds Per Read

< 0.005	excellent
0.005 – 0.010	good
0.010 – 0.015	fair
> 0.015	bad

Average Seconds Per Write

< 0.001	excellent
0.001 – 0.002	good
0.002 – 0.004	fair
> 0.004	bad

Use the include and exclude filters to control which drives to display. Do not include the colons. Normally you would set one or the other, not both. But if you can find a use case for both then go

ahead. The two filters are evaluated as include first then exclude, whatever passes both is displayed.

To display only drive C and D:

```
include:CD,exclude:,
```

To display all drives except F: and G:

```
include:,exclude:FG,
```

To display all drives except A: and B:, don't specify either as this is the default. It would look like this:

```
include:,exclude:ab,
```

To display all drives including drive B: but not drive A:

```
include:,exclude:A,
```

To display all drives including A: and B:

```
include:,exclude:,
```

By default, drive performance data is retrieved even if not displayed. If you want to stop retrieving performance data to save a few cpu cycles, set the diskio option to 0.

```
diskio:0,
```

To limit the maximum number of drives displayed, you can set the count option to the desired maximum number.

NETCONNECTIONS

Shows the current number of network connections.

Return Values:

%1	number of connections
----	-----------------------

NETPACKETS

Shows the total network ipv4 packs sent and received. Use the mtu option to convert packets to bytes.

Available Options:

mtu	multiply by this number to get bytes
-----	--------------------------------------

Return Values:

%1	packets (or bytes) received
%2	packets (or bytes) sent

NETPACKETSRATE

Shows the current ipv4 send and receive rate of network packets. Use the mtu option to convert packets to bytes.

Available Options:

mtu	multiply by this number to get bytes
-----	--------------------------------------

Return Values:

%1	packets (or bytes) received / sec
%2	packets (or bytes) sent / sec

NETWORKINTERFACES

Displays detailed information for the selected network interfaces.

Where a network interface card has multiple ports and hence is connected to multiple networks, this will show the first IP address, mask and prefix length for ipv4 and ipv6. To show all addresses, use the ipv4list and ipv6list options in conjunction with return values %25 and %26.

Available Options:

operstatus	show only interfaces that have the specified operational status (1=connected, 2=active not connected)
iftype	show only interfaces of the specified interface type
friendlyfilter	filter on the friendly name
descfilter	filter on the description
ipv4list	display template for return value %25
ipv6list	display template for return value %26

Return Values:

%1	adapter name
%2	friendly name
%3	description
%4	mac address
%5	mtu
%6	interface type
%7	operational status
%8	first ipv4 address
%9	first ipv4 mask
%10	first ipv4 prefix length
%11	first ipv6 address
%12	first ipv6 mask (not implemented)
%13	first ipv6 prefix length
%14	first lease expiry
%15	ipv4 dns servers (space delimited list)
%16	ipv6 dns servers (space delimited list)
%17	ipv4 gateways (space delimited list) *
%18	ipv6 gateways (space delimited list) *

%19	ipv4 dhcp server *
%20	ipv6 dhcp server *
%21	ipv4 wins servers (space delimited list) *
%22	ipv6 wins servers (space delimited list) *
%23	transmit link speed *
%24	receive link speed *
%25	ipv4 all addresses (see explanation below)
%26	ipv6 all addresses (see explanation below)

* These values are not available on Windows 2000 and XP.

The `operstatus` and `iftype` filters can contain multiple values separated by a space. For example, to show ethernet and wireless interfaces, "`iftype:6 71`". To show only active interfaces, "`operstatus:1`".

Common Interface Types:

1	Other
6	Ethernet
9	Token ring
23	PPP (VPN)
24	Software loopback
37	ATM
71	IEEE 802.11 wireless (wifi)
131	Tunnel type encapsulation
144	IEEE 1394 Firewire
243	Cellular

The *friendlyfilter* and *descfilter* options may contain *includes* denoted by a '+' sign and *excludes* denoted by a '-' sign.

Example:

```
friendlyfilter:+wireless-virtual
```

Shows all interfaces that contain the keyword "wireless" and will exclude interfaces that contain the keyword "virtual".

If there is more than one filter match, they are evaluated left to right and the last one wins. For example, "`friendlyfilter:+wireless-wire`" will include all interfaces with "wireless" in the friendly name and then exclude any interface with "wire" in the friendly name, resulting in no wireless interfaces.

An interface must match all the given filters in order to be displayed.

Network Interfaces With Multiple Ports

If a network interface card has multiple ports and hence connected to multiple networks, it will have multiple IP addresses, masks and prefix lengths. In this case, return values %8, %9 and %10 will

return the first ipv4 address for the card and return values %11, %12 and %13 will return the first ipv6 address for the card.

If you want to display all addresses for the card, add the %25 and/or %26 return values to the display template. By default this will show all ipv4 and ipv6 addresses respectively, separated by line feeds.

You can modify the format of %25 by adding the ipv4list option and you can modify the format of %26 by adding the ipv6list option. For the ipv4list option you can include the %8, %9 and %10 return values. For the ipv6list option you can include the %11, %12 and %13 return values. For example:

```
ipv4list:%8/%10|,ipv6list:%11|,display:%25|%26
```

Will show all ipv4 addresses and prefix lengths in %25 and just the ipv6 address in %26.

You can include the pipe symbol to indicate a new line or a comma or just a space if you prefer. The final pipe, comma or space will be trimmed from the end of the list so you don't have a blank line in the display.

OEMINFO

Reads the OEM Information set in either the OEMInfo.ini file or the OEMInformation registry key in either the 64 bit or 32 bit branch of the registry.

Return Values:

%1	manufacturer
%2	model
%3	Support Provider
%4	Support URL
%5	Support Hours
%6	Support Phone

OSBUILD

Shows the Windows operating system version and build number.

Return Values:

%1	build number
----	--------------

PAGEFAULTS

Shows both total and hard page faults (with reads and writes). The hard page faults is usually the one you're interested in. This shows page file activity which is what can potentially kill system

performance. The hard page faults are further broken down into ‘pages read from disk per second’ and ‘pages written to disk per second’.

The *read hit ratio* is the percentage of page reads found in memory. A value of over 80% is good. The *read miss ratio* is the percentage of page reads that had to go to disk. A value of over 80% is bad. This means the page file on the hard disk is being thrashed and system performance may be suffering. If this is a regular occurrence the system could probably use more physical RAM. A threshold on either of the hit ratios would be useful.

Return Values:

%1	total faults per second
%2	hard page faults per second
%3	pages read from disk per second
%4	pages written to disk per second
%5	read hit ratio
%6	read miss ratio

PAGEFILE

Shows Windows page file usage.

Return Values:

%1	page file used
%2	total page file size
%3	percent page file used
%4	page file free
%5	percent page file free

PHYSICALDISKS

Shows information about physical disks including manufacturer, size, partitions, logical disks etc.

Return Values:

%1	name
%2	bytes per sector
%3	caption
%4	description
%5	device id (in the form of \\.\PHYSICALDRIVE0)
%6	index
%7	physical interface type
%8	list of logical disks on this physical disk and the partition on which it resides this is a line feed delimited list, one logical disk per line
%9	media type
%10	manufacturer's model
%11	number of partitions

%12	list of partitions on this disk and any logical disks in them this is a line feed delimited list, one partition per line
%13	manufacturer's serial number
%14	size in bytes
%15	status string

This item is available on Windows Vista and later and Windows Server 2008 and later.

PHYSICALRAM

Shows physical memory usage.

Return Values:

%1	used ram
%2	total ram
%3	percent used ram
%4	free ram
%5	percent free ram

PROCESSCOUNT

Total number of Windows processes and threads.

Return Values:

%1	total processes
%2	total threads *

* Total threads requires DesktopInfo64.

PROCESSMEM

Shows memory and cpu usage for the given process. If there is more than one instance of the given process name, they are all included in the total.

Available Options:

key	name of the process
-----	---------------------

Return Values:

%1	working set size
%2	page file usage
%3	page fault count
%4	cpu usage
%5	commit size *
%6	process handle count
%7	process thread count *

* Requires desktopinfo64. To enable this, set the *desktopinfo64* option in the [options] section:
desktopinfo64=1

PROXY

Shows network proxy server and enabled status.

Return Values:

%1	proxy server
%2	enabled state

RDS

Shows assorted information about Terminal Server / Remote Desktop Services.

Return Values:

%1	total session count
%2	active console session id the session that is currently attached to the physical console
%3	current session id the remote session in which DTI is running
%4	initial program Initial program that Remote Desktop Services runs when the user logs on. I'm not sure if this is still in use.
%5	application name The published name of the application that the session is running. I'm not sure if this is still in use.
%6	current session user name
%7	current session host name
%8	current session host address (ipv6 or ipv4, see notes below)
%9	session address ipv4 the address assigned to the session, if one is assigned

The client network address is reported by the RDP client itself when it connects to the server. This could be different than the address that actually connected to the server. For example, suppose there is a NAT between the client and the server. The client can report its own IP address, but the IP address that actually connects to the server is the NAT address. For VPN connections, the IP address might not be discoverable by the client. If it cannot be discovered, the client can report the only IP address it has, which may be the ISP assigned address. Because the address may not be the actual network address, it should not be used as a form of client authentication.

https://docs.microsoft.com/en-us/windows/win32/api/wtsapi32/ns-wtsapi32-wts_client_address

Further, if the client has ipv6 enabled, it may report this address in preference to the ipv4 address. If you really want to see the ipv4 address in the RDS item, you will need to disable ipv6 on the connecting client.

See the *desktopinfo-advanced.ini* file for a comprehensive example.

RDSSESSIONS

Displays a list of Terminal Server / Remote Desktop Services sessions.

Available Options:

filter	filter on the station name (session name)
filter-state	filter on the session state

Return Values:

%1	station name (session name)
%2	session id
%3	session state

This returns a number. To convert this to something more meaningful, use a lookup table.

The *filter* option may contain *includes* denoted by a '+' sign and *excludes* denoted by a '-' sign.

Example: Include only RDP sessions.

```
filter:+RDP
```

The *filter-state* option works with the state value and may contain *includes* denoted by a '+' sign and *excludes* denoted by a '-' sign.

Example: Include only active sessions.

```
filter-state:+0
```

Example:

```
RDSSESSIONS=interval:20,row-text:%1,display:sid:%2 state:%rds_state_%3%
```

This displays the session name in the left column and the session id and state in the right column. The state uses a lookup table to translate the value into a description.

See the *desktopinfo-advanced.ini* file for a comprehensive example.

Session State Values

0	Active
1	Connected
2	ConnectQuery
3	Shadow
4	Disconnected
5	Idle
6	Listen
7	Reset
8	Down
9	Init

REG

Shows the given value from the registry. Not to be confused with the REGISTRY monitor item, this item just retrieves the given value at a regular interval. It can only show integers and strings.

Available Options:

value	the registry key and value name to be read
wow	32=read from the 32 bit registry 64=read from the 64 bit registry
host	optional remote host

Return Values:

%1	the value retrieved from the registry
----	---------------------------------------

The value option should contain the root key as well as the key and value name:

```
value:HKEY_LOCAL_MACHINE\SOFTWARE\TeamViewer\Version5\ClientID
```

To access the registry on a remote host, set the host option to the host name of the remote computer. Both computers must have the Remote Registry service running. The Desktop Info user obviously must have permissions on the remote host. Remote registry access may be restricted by domain or local policy. For more information on a possible solution, go to:

<https://support.microsoft.com/en-au/help/951016/description-of-user-account-control-and-remote-restrictions-in-windows>

For more information on the wow option, go to:

<https://docs.microsoft.com/en-us/windows/desktop/winprog64/accessing-an-alternate-registry-view>

REGEXIST

Shows whether the given registry key or value exists. Use the boolean formatting to show a meaningful display.

Available Options:

value	the registry key or key\value to be tested
wow	32=read from the 32 bit registry 64=read from the 64 bit registry
host	optional remote host

Return Values:

%1	true or false if the key or key\value exists
----	--

The value option should contain the root key as well as the key and optional value name:

```
value:HKEY_LOCAL_MACHINE\SOFTWARE\TeamViewer\Version5\  
ClientID,display:%1[b:Yes:No]
```

To access the registry on a remote host, set the host option to the host name of the remote computer. Both computers must have the Remote Registry service running. The Desktop Info user obviously must have permissions on the remote host. Remote registry access may be restricted by domain or local policy. For more information on a possible solution, go to:

<https://support.microsoft.com/en-au/help/951016/description-of-user-account-control-and-remote-restrictions-in-windows>

REGISTRY

Monitors registry keys or values for changes. User variables may be used in the key or value to monitor.

Available Options:

tree	0=monitor only the key (default) 1=monitor the tree starting at key
key	the key or key\value to monitor

Return Values:

%1	required value
%2	date/time of change

If you specify a key, %1 is not used. You can also choose to monitor just that key or the entire tree starting at that key.

The following example will monitor a value in the Run key.

```
REGISTRY=text:Run Key Test,key:HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\Test
```

If the registry key contains a comma, precede it with a backslash.

Invalid Key

```
HKEY_LOCAL_MACHINE\SOFTWARE\Apple Computer, Inc.
```

Valid Key

```
HKEY_LOCAL_MACHINE\SOFTWARE\Apple Computer\, Inc.
```

SERVICEPACK

Shows the Windows service pack edition.

Return Values:

%1	service pack
----	--------------

SET

Sets the value for the given user variable. See the section on [User Variables](#). It is hidden by default.

Available Options:

key	the user variable you want to set
value	the value to place in the user variable

The SET item can be inserted anywhere in the [items] section. It is unique among items in that it is independent of pages. It will always be evaluated regardless of the currently displayed page. Therefore, the only purpose of the *active* option is to declare the page upon which it would be visible if the *hidden* option is set to zero.

The simplest form of the SET item is:

```
SET=key:white,value:#ffffff
```

This will make the user variable available to all items in the configuration.

Color Variables

You can use variables to create your own preset colors to be used throughout the config. For example:

```
[items]
SET=key:white,value:#ffffff
SET=key:blue,value:#ff0000
PAGE=1
HOST=color:%white%,display:%1
USER=color:%blue%,display:%1
```


If you want to debug the SET item, set the *hidden* option to zero and *active* to any page:

```
SET=key:blue,value:#ff0000,hidden:0
```

SNAPSHOT

Creates an image snapshot of the current display. You can create a bare image of any size containing the DTI display or supply a jpg/png image file as a background with the DTI display placed on top.

Available Options:

width	width of the bare image (default is DTI width)
height	height of the bare image (default is DTI height)
bgcolor	background color of the bare image (default is DTI formcolor)
top	vertical placement of the DTI display within the image
left	horizontal placement of the DTI display within the image
infile	path and file name of a jpg/png image to use as the background
outfile	path and file name of the resulting output jpg/png image

A bare image is just a blank canvas of any color with the DTI display placed in any location. Use the width and height options to set the size of the output image and the bgcolor option to set the background color. Use the top and left options to locate the DTI display within the output image.

To use an existing jpg/png image as the background, use the infile option. Use the top and left options to locate the DTI display on that image. Width, height and bgcolor are not used, the output image is the same size as the background image.

The input background file can be a jpg or png file. The resulting output file can be jpg or png.

TEXT

Displays any fixed key/value text. Useful for any kind of internal identification or other static information such as a machine's context, use, operator, tech support info. It can also be used to display any number of user variables on one line.

Available Options:

text	Text to display in the left column
display	Text to display in the right column

TIMEZONE

Current Windows time zone.

Return Values:

%1 time zone

TOPPROCESSCPU

Shows the processes with the highest cpu usage. Desktop Info can only show processes it has permissions to read. Running as administrator or a member of the administrators group is enough for most processes but not for some system processes. Desktop Info will attempt to enable privileges to read system process information.

Available Options:

maxrows	number of top processes to display (default=1)
filter	include or exclude processes based on process name

Return Values:

%1	process name
%2	process id
%3	percent

The filter option may contain *includes* denoted by a '+' sign and *excludes* denoted by a '-' sign.

Example:

```
filter:+svchost
```

Shows only processes that contain "svchost" in the process name.

If there is more than one filter match, they are evaluated left to right and the last one wins. For example, "filter:+svchost.exe-host" will include all processes with "svchost.exe" and then exclude any process with "host", resulting in no return results.

A process name must match all the given filters in order to be displayed.

TOPPROCESSMEM

Shows the process with the highest memory usage. The memory counter used can be either the "Working Set Size" or "Page File / Private Usage" (Microsoft sometimes refers to the latter as "Commit Charge" or "Commit Size"). If desktopinfo64.exe is running, 'pu' will return Private Usage, otherwise it will return Page File Usage. In theory they are the same value., however Microsoft states that on Windows 7 and earlier, Page File Usage is always 0. If desktopinfo64.exe is not running, Private Usage is not available and Page File Usage is returned.

Available Options:

counter	which memory counter to use
	ws = Working Set Size

pu = Page File Usage (32 bit mode)

or Private Usage (64 bit mode)

maxrows number of top processes to display (default=1)

up to 10 in 64 bit mode

filter include or exclude processes based on process name

Return Values:

%1 process name

%2 process id

%3 usage bytes

The filter option may contain *includes* denoted by a '+' sign and *excludes* denoted by a '-' sign.

Example:

```
filter:+svchost
```

Shows only processes that contain "svchost" in the process name.

If there is more than one filter match, they are evaluated left to right and the last one wins. For example, "filter:+svchost.exe-host" will include all processes with "svchost.exe" and then exclude any process with "host", resulting in no return results.

A process name must match all the given filters in order to be displayed.

TOPPROCESSPF

Shows the processes with the highest page faults. This is not page faults per second, it's the total page faults since the process started.

Available Options:

maxrows number of top processes to display (default=1)

filter include or exclude processes based on process name

Return Values:

%1 process name

%2 process id

%3 page faults

The filter option may contain *includes* denoted by a '+' sign and *excludes* denoted by a '-' sign.

Example:

```
filter:+svchost
```

Shows only processes that contain "svchost" in the process name.

If there is more than one filter match, they are evaluated left to right and the last one wins. For example, "filter:+svchost.exe-host" will include all processes with "svchost.exe" and then exclude any process with "host", resulting in no return results.

A process name must match all the given filters in order to be displayed.

UNREADMAIL

Show the number of unread mails.

Return Values:

%1	number of unread mails
----	------------------------

UPTIME

Total time Windows has been running.

Return Values:

%1	days
%2	hours
%3	minutes
%4	seconds
%5	total up time in seconds

Use the total up time in seconds (%5) to do useful calculations.

USER

Current Windows user account name.

Return Values:

%1	user name
----	-----------

VIRTUALMEMORY

Shows Windows virtual memory usage.

Return Values:

%1	used memory
%2	total memory
%3	percent used memory

%4	free memory
%5	percent free memory

WMI

Executes a custom WMI query. Because there is an implied "select * from ", all properties are returned so you can include as many properties as you like in the display output.

Available Options:

namespace	WMI namespace such as 'root\wmi' or 'root\cimv2'
query	the wmi class and optional where clause but NOT the select clause (this is implied)
maxrows	the maximum number of rows to display There will always be at least one row (unless hide-no-result is set) so 0 and 1 are effectively the same thing.

Return Values:

Return values are indicated by enclosing a wmi property name in % signs.
eg a bios property, SerialNumber, is indicated as %SerialNumber%. This is replaced in the display template by the actual value returned by the WMI query.
Additionally, the %rowcount% property will display the number of rows in the WMI result set. See below for an example.

Example:

```
WMI=namespace:root\cimv2,query:Win32_Bios,display:%SerialNumber%
```

WMI Where Clause

This one displays multiple values from the wmi query with added text. Notice that the *where* clause contains a literal string in double quotes. Notice also how the *where* clause may include the 'like' keyword with wildcards.

```
WMI=namespace:root\cimv2,query:Win32_PerfFormattedData_Tcpip_NetworkInterface where Name like "%Wireless%",display:Rcvd: %BytesReceivedPersec% Sent: %BytesSentPersec% bytes per sec
```

You can add the NOT key to exclude rows from the result. It goes before the property in question thus:

```
query:Win32_PerfFormattedData_Tcpip_NetworkInterface where not Name like "%Wireless%"
```

In this example, WMI doesn't have a single query that returns the screen resolution and color depth so we run two queries and combine the output. The first query returns the color depth and stores it in a user variable (see below). The second query returns the resolution and displays it along with the previous user variable.

```
WMI=namespace:root\cimv2,  
query:Win32_DisplayConfiguration,set:BitsPerPel,display:%BitsPerPel%  
WMI=namespace:root\cimv2,query:Win32_DesktopMonitor, display:  
%ScreenWidth%x%ScreenHeight%x%BitsPerPel%
```

You can also append the number formatting as described earlier. You should make sure the selected format matches the type of value you are trying to format.

User Variables

You can use user variables in a WMI query. This allows you to do lookups and other interesting things using information already retrieved. For example:

```
USER=set:username  
WMI=namespace:root\cimv2,query:Win32_NetworkLoginProfile where Caption  
like "%username%",display:%Caption% %LastLogon%
```

WMI Row Count

The *maxrows* option will display at most the given number of rows. If you set *maxrows* to 1 and set the *display* option to *%rowcount%*, you can show just the number of rows without displaying all the rows. This is useful, for example, for displaying the number of Windows updates available without showing all the updates.

```
WMI=interval:60,text:Windows Updates,namespace:root\cimv2,  
query:win32_quickfixengineering,maxrows:1,display:%rowcount% updates  
available
```

WMI Date/Time Formatting

If a CIM_DATETIME field is detected, it is automatically split into its component parts and added to the result set. These new date/time component properties are named starting with the original property name, followed by an underscore, followed by the date/time component.

```
_year, _month, _day, _hour, _minute, _second, _msecond, _utcoffset
```

See the Microsoft page for more information:

<https://docs.microsoft.com/en-us/windows/desktop/wmisdk/cim-datetime>

These components are available in the [display template](#) with date/time formatting so you can create a more user friendly display.

For example, the Win32_OperatingSystem query contains a property called *%InstallDate%*. This property is displayed as "20181217091440.000000+120". It's readable, kind of. It is a CIM_DATETIME type and so it will be automatically split into its component parts with the result

that, along with the %installdate% property there is also %installdate_year%, %installdate_month%, %installdate_day% etc. These new properties are available to the display template and may be formatted to your taste. For example:

```
text:Windows Install Date, display:%installdate_day%
%installdate_month%[mmm] %installdate_year%
```

The CIM_DATETIME field has a very specific format. Not all WMI date/time fields are of this type. Only this type will be split.

WMI Explorer is an easy tool to explore the WMI system. <http://www.ks-soft.net/>

Useful WMI Queries

Here are some useful WMI queries that you can plug straight into the desktopinfo.ini file.

Anti Malware Service

```
WMI=interval:60,text:Anti Malware Service, namespace:root\cimv2,
query:Win32_Service where name="MsMpSvc",display:%state%\, %status%
```

Anti Virus Product

```
WMI=interval:60,text:Anti-Virus,namespace:root\
SecurityCenter2,query:AntiVirusProduct,display:%displayName%
%productState%[bit:4096:Enabled:Disabled] and %productState%
[bit:393216:up to date:not up to date]
```

Asset Tag

```
WMI=text:Asset Tag, namespace:root\cimv2,
query:Win32_SystemEnclosure,display:%SMBIOSAssetTag%
```

Batteries

```
WMI=interval:60,text:Batteries,namespace:root\
cimv2,query:Win32_Battery,row-text:%Description%,display:
%EstimatedChargeRemaining%
```

Battery

This includes the battery status message in brackets after the estimated charge remaining. Note the WMI line should all be one line in your ini file.

```
set=key:battery1,value:Disconnected
set=key:battery2,value:Connected
WMI=interval:60,text:Battery,namespace:root\cimv2,
query:Win32_Battery,display:%EstimatedChargeRemaining%% (%battery
%BatteryStatus%%)
```

To add a bar chart, add the following:

```
chart:bar2 scale:linear max:100 series1:%EstimatedChargeRemaining%
threshold:-10
```

Countdown

This is a simple countdown starting from boot time to a number of seconds in the future. In this example, 604800 seconds is seven days. Change the value in the first line to your desired number of seconds.

```
set=key:t,value:604800
UPTIME=interval:1,text:Countdown,display:{{trunc((%t%-%5)/86400)}}d
{{trunc(mod((%t%-%5)\,86400)/3600)}}h
{{trunc(mod((%t%-%5)\,3600)/60)}}m {{mod(mod((%t%-%5)\,3600)\,60)}}s
```


Cpu Temperature

The formula for Fahrenheit is $(\%temperature\%-273.15)*9/5+32$.

```
WMI=interval:5,text:Cpu Temp,namespace:root\
cimv2,query:Win32_PerfFormattedData_Counters_ThermalZoneInformation,di
splay: {{1.1f:%temperature%-273.15}}C
```

Cpu Temperature

This uses a different wmi class. If one doesn't work for you, maybe the other one will. This displays the result in Celsius.

```
WMI=interval:5,text:Cpu Temp,namespace:root\
wmi,query:MSAcpi_ThermalZoneTemperature,display: {{1.1f:
%CurrentTemperature%/10-273.15}}C
```

Ethernet Nic Speed

Set the 'where' clause so it catches your adapter.

```
WMI=interval:5,text:Ethernet Speed, namespace:root\cimv2,
query:Win32_NetworkAdapter where Name like "%PCI-E%",display:%Speed
%[1.0B]Bps
```

Lanman Service

```
WMI=interval:60,text:Lanman Service, namespace:root\cimv2,
query:Win32_Service where name="lanmanserver",display:%state%,
%status%
```

Logical Disks

```
WMI=interval:10,text:Logical Disks, namespace:root\cimv2,
query:Win32_PerfFormattedData_PerfDisk_LogicalDisk where Name !=
"_Total",display:%name% r:%DiskReadBytesPersec%[3.1k]KiBps\, w:
%DiskWriteBytesPersec%[3.1k]KiBps
```

Network Latency

```
WMI=interval:60,text:Network Latency, namespace:root\cimv2,
query:Win32_PingStatus where address='1.1.1.1',display:%ResponseTime
%ms (Cloudflare)
```

Open Hardware Monitor

Open Hardware Monitor is a free open source software that monitors temperature sensors, fan speeds, voltages, load and clock speeds of a computer. When it is running, it provides a couple of WMI classes that can be used to retrieve this information. You can download it from <https://openhwaremonitor.org/>

```
COMMENT=text:Open Hardware Monitor - Hardware,style:u
WMI=interval:10,text:Hardware,namespace:root\
OpenHardwareMonitor,query:Hardware,row-text:Name|Type / Id,display:
%Name%|%HardwareType% / %Identifier%
COMMENT=text:Open Hardware Monitor - Sensor,style:u
WMI=interval:10,text:Hardware,namespace:root\
OpenHardwareMonitor,query:Sensor where SensorType = "Temperature",row-
text:%Identifier%,display:%Value%C
```

Physical Disk Status

```
WMI=interval:60,text:Physical Disks,namespace:root\
cimv2,query:Win32_DiskDrive,display:%Caption%| DeviceID: %DeviceID%|
Model:%Model%| Status:%Status%
```

Printers (All)

```
WMI=text:Printers,namespace:root\cimv2,query:Win32_Printer,display:
%Name% (%Status%)
```

Printer (Default)

```
WMI=text:Default Printer, namespace:root\cimv2,query:Win32_Printer
where default = "True",display:%Name% (%Status%)
```

Server Sessions

```
WMI=interval:60,text:Server Sessions, namespace:root\cimv2,
query:Win32_PerfFormattedData_PerfNet_Server,display:Sessions:
%ServerSessions%\, Files: %FilesOpen%
```

Shared Resources

```
WMI=interval:60,text:Shared Resources, namespace:root\cimv2,
query:Win32_Share where not Name like "%$",display:%Name% [%Status%]
```

Tcpv4 Connections

```
WMI=interval:10,text:Tcpv4 Connections, namespace:root\cimv2,
query:Win32_PerfFormattedData_Tcpip_TCPv4,display:
%ConnectionsEstablished%
```

Tcpv6 Connections

```
WMI=interval:10,text:Tcpv6 Connections, namespace:root\cimv2,
query:Win32_PerfFormattedData_Tcpip_TCPv6, display:
%ConnectionsEstablished%
```

Video Controller

```
WMI=text:Video Controller, namespace:root\cimv2,
query:Win32_VideoController,display:%Caption%
```

Windows Workgroup

```
WMI=text:Workgroup,namespace:root\
cimv2,query:Win32_ComputerSystem,display:%Domain%
```

Windows Update Service

```
WMI=interval:60,text:Windows Upd Service, namespace:root\cimv2,
query:Win32_Service where name="wuauserv",display:%state%\, %status%
```

Windows Firewall

```
WMI=interval:60,text:Windows Firewall, namespace:root\cimv2,
query:Win32_Service where name="mpssvc",display:%state%\, %status%
```

Wifi Speed

```
WMI=interval:5,text:Wifi Speed, namespace:root\cimv2,
query:Win32_NetworkAdapter where Name like "%wireless%",display:%Speed
%[1.0B]Bps
```

Useful Windows Tools

Here are some useful Windows tools you can add to Desktop Info controls.

Certificate Manager

```
CONTROL=text:,display:Certificate Manager,uri:certmgr.msc
```

Computer Management

```
CONTROL=text:,display:Computer Management,uri:compmgmt.msc
```

Control Panel

```
CONTROL=text:,display:Control Panel,uri:control
```

Device Manager

```
CONTROL=text:,display:Device Manager,uri:devmgmt.msc
```

Devices and Printers

```
CONTROL=text:,display:Devices and Printers,uri:%SystemRoot%\System32\control.exe,args:printers
```

Disk Management

```
CONTROL=text:,display:Disk Management,uri:diskmgmt.msc
```

Local Certificate Manager

```
CONTROL=text:,display:Local Certificate Manager,uri:certlm.msc
```

Local Group Policy Editor

```
CONTROL=text:,display:Local Group Policy,uri:gpedit.msc
```

Local Users and Groups

```
CONTROL=text:,display:Local Users and Groups,uri:lusrmgr.msc
```

Network Connections

```
CONTROL=text:,display:Network Connections,uri:%SystemRoot%\System32\control.exe,args:ncpa.cpl
```

Programs and Features

```
CONTROL=text:,display:Programs and Features,uri:%SystemRoot%\System32\control.exe,args:appwiz.cpl
```

Reliability Monitor

```
CONTROL=text:,display:Reliability Monitor,uri:control,args:/name Microsoft.ActionCenter /page pageReliabilityView
```

Resource Monitor

```
CONTROL=text:,display:ResourceMonitor,uri:resmon
```

System Information

```
CONTROL=text:,display:System Info,uri:msinfo32.exe
```

System Properties

```
CONTROL=text:,display:System Properties,uri:%SystemRoot%\System32\
control.exe,args:sysdm.cpl
```

Windows Admin Tools

```
CONTROL=text:,display:Win Admin Tools,uri:C:\ProgramData\Microsoft\
Windows\Start Menu\Programs\Administrative Tools
```

Windows Updates

```
CONTROL=text:,display:Windows Updates,uri:control,args:/name
Microsoft.WindowsUpdate
```

References

WMI Explorer

<http://www.ks-soft.net/>

Core Temp

<http://www.alcpu.com/CoreTemp/>

Understanding Kilobytes

<https://en.wikipedia.org/wiki/Kilobyte>

Network Interfaces Reference

https://docs.microsoft.com/en-us/windows/desktop/api/iptypes/ns-iptypes-ip_adapter_addresses_lh

The Registry Wow View

<https://docs.microsoft.com/en-us/windows/desktop/winprog64/accessing-an-alternate-registry-view>

URI: The Universal Resource Identifier

<https://www.w3.org/wiki/URI>

Open Hardware Monitor

<https://openhardwaremonitor.org/>

Code Page Identifiers

<https://docs.microsoft.com/en-us/windows/win32/intl/code-page-identifiers>

Release Notes

Version v0.1.0

2005

Limited distribution test release.

Version v0.1.1

2005

Change 1: Fixed the nasty flicker some systems were getting.

Change 2: Fixed the odd characters after the domain name.

Change 3: Less full refreshes means less cpu time.

Change 4: Fixed disk figures for very small drives.

Version v0.2.0

2005

Change 1: Added time zone info.

Change 2: Some optimisations.

Change 3: Added refresh intervals.

Change 4: Added domain controller info.

Change 5: Added event logs.

Version v0.2.1

December 2005

Change 1: Added terminal server session count.

Change 2: Fixed some stuff in event logs.

Version v0.2.2

December 2005

Change 1: Added DirectX version.

Change 2: Adjusted domain controller. I can't directly test this one.

Change 3: Added network packet stats and rates.

Change 4: Added network connection count.

Change 5: Added double click refresh.

Version v0.3.0

Change 1: Fixed a bunch of memory issues.

Change 2: Fixed Terminal Services sessions and added session list.

Change 3: Added auto font size.

Change 4: Added 'missing ini' default values.

Change 5: Implemented proper ini file monitoring.

Change 6: Added screen info.

Change 7: Fixed display for 256 color remote desktop.

Change 8: Added files monitor options.

Version v0.3.1

June 2006

Desktop Info - Manual

Change 1: Added Up Time.

Change 2: Added support for cpu times for Windows 2000.

Change 3: Fixed Domain Controller

Change 4: Fixed multiples of same file watch type bug.

Change 5: Fixed memory sizes over 2GB.

Version v0.4.0

July 2006

Change 1: Added top process cpu time.

Change 2: Added top process memory usage.

Change 3: Added file version watch type.

Change 4: Added registry watch types.

Change 5: Added cpu count.

Change 6: Additional checking for terminal sessions.

Version v0.4.1

December 2006

Change 1: Added unread mail.

Change 2: Added formright and formbottom options.

Version v0.4.2

October 2007

Change 1: Increased fixed disks to eight.

Change 2: Added multiple ip addresses.

Change 3: Fixed problem with missing ini file.

Change 4: Split network gateway entries to separate lines.

Change 5: Added percentages to memory and disk displays.

Version v0.5.0

July 2008

Change 1: I think I've nailed the show desktop thing.

Version v0.5.1

Change 1: Fixed problem with some USB drives.

Change 2: Solved refresh problem when removable drives come and go.

Version v0.6.0

November 2008

Change 1: Reworked the file monitoring so it works on local and remote files and folders. You'll need to adjust your ini file entries as noted above.

Change 2: Reworked the registry monitoring to merge the two types. You'll need to adjust your ini file entries as noted above.

Change 3: Added the /f command line option to show a visible, moveable form.

Change 4: Added the /a command line option to show all items regardless of the ini file settings.

Change 5: Added right click context menu.

Change 6: The network entries are grouped so all information for each adapter is together.

Change 7: Tested on a variety of Windows platforms. The results are noted above.

Version v0.7.0

January 2009

Change 1: Added Disk IO on fixed logical drives.

Change 2: Added disk queue length.

Change 3: Changed ini file format as noted above. Will continue to read the old format while the Intervals section exists.

Change 4: Added item colors.

Change 5: Fixed bug in Top Process Cpu

Change 6: Fixed bug in cpu times.

Change 7: Added charts.

Change 8: Added thresholds.

Change 9: Added exception handlers to process enumerators.

Version v0.7.1

Change 1: Fixed an access violation in the disk io routines.

Change 2: Fixed refresh when disks come and go.

Version v0.8.0

Change 1: Added colors to the file and registry monitors.

Change 2: Added option to disable the context menu.

Change 3: Added option to toggle the indents on disks/networks.

Change 4: Modified registry monitor to optionally show values.

Change 5: Registry item names are shown as found in the ini file.

Change 6: Added event log monitor and removed redundant event logs from the items.

Version v0.8.1

Change 1: Improved disk change handling.

Change 2: Fixed file monitor where file/directory does not exist or disappears or reappears.

Change 3: Fixed divide by zero error in fixed disks.

Version v0.9.0

Change 1: Refactored the chart display code to be more useful and added charts to more items. See the charts section for more information.

Change 2: Refactored form display code. Options have changed accordingly. Auto form width, auto font size and font color are gone. You should make sure every item has the color set. See the options section for more information.

Change 3: Rewritten all the network adapter stuff. Added change handling. Added filtering. See the options section for more information.

Change 4: Prevent form from resetting it's position after it is dragged.

Change 5: Rewrote the bar chart.

Change 6: Fixed problem with event log monitor thread not terminating correctly.

Change 7: Added the following items: printer, printerstatus, displaycontroller, bios, motherboard, workgroup. Printers is work in progress.

Change 8: Added language file.

Change 9: Added screen saver option.

Change 10: Added msnstatus option.

Change 11: Changed method for retrieving service pack.

Version v1.0.0

Change 1: Added page faults and top process page faults.

Desktop Info - Manual

Change 2: The order of displayed items now follows the order in the ini file.

Change 3: New ini file item format. This will make it easier to read and easier for me to add new options. See the item section above for more details.

Change 4: Fixed the process name for all known varieties of Windows. If a process name can't be retrieved for any reason, it will display <n/a> and maybe an error message.

Change 5: Added cpu temperature from wmi. See the miscellaneous section above.

Change 6: Fixed line chart width in screen saver mode.

Change 7: Subdued some redundant refreshes.

Change 8: Some memory usage optimisations.

Change 9: Fixed display controller and bios on Windows 7 and hopefully haven't broken it elsewhere.

Change 10: Added oeminfo. See miscellaneous section above.

Change 11: Added item font style.

Version v1.0.1

June 2010

Change 1: Fixed fatal crash on startup as the result of an access denied problem when retrieving OS info in limited access Windows account.

Version v1.1.0

January 2011

Change 1: Added battery status.

Change 2: Added option to disable ClearType.

Change 3: Added header item. See ini file.

Change 4: Added cpu kernel time.

Change 5: Added inimonitor time to options. This is the number of seconds to check the desktopinfo.ini file for changes.

Change 6: Added osbuild.

Change 7: Fixed a problem with reading process information for some system processes.

Change 8: Removed DISKQUEUE item. It's included on the DISKIO item.

Change 9: Added filter option to FIXEDDISK item.

Version v1.1.1

Change 1: Changed the way the Windows architecture is determined.

Change 2: Fixed the transparency on 16 bit color display.

Version v1.1.2

Change 1: Forgot to switch off debug causing large log file, oops.

Version v1.2.0

October 2013

Change 1: Fixed index out of bounds when battery chart type is 2.

Change 2: Added network PROXY item.

Change 3: Changed HEADER item into more general COMMENT item.

Change 4: COMMENT is ignored when calculating width of column 1 allowing it to display over both columns.

Change 5: Added underline to style option.

Change 6: Changed FORMCOLOR in the main options section to a bgr type value to keep it consistent with other colors. 000000 means transparent in normal mode or black background in form mode.

Change 7: Converted file monitor into a regular item. Now you can put one or more file monitor items anywhere in the items list.

Desktop Info - Manual

Change 8: Converted registry monitor into a regular item. Now you can put one or more registry monitor items anywhere in the items list.

Change 9: Fixed problem with monitoring registry on 64 bit Windows.

Change 10: Converted event log monitor into a regular item. Now you can put one or more event log monitor items anywhere in the items list.

Change 11: Added AUDIOCONTROLLER item.

Change 12: Added logging option.

Change 13: Added /ini command line option.

Change 14: Added SERIALNUMBER item.

Change 15: Added multiple IP addresses per nic.

Change 16: Added option to allow/disallow dragging the form.

Change 17: Added offset property to network items.

Change 18: Added multi core cpu item, CPUUSAGE.

Change 19: Added activeonly property to NETWORKADAPTER.

Change 20: Added count property to control maximum number of multi items displayed (CPUUSAGE, NETWORKADAPTER, FIXEDDISK, PRINTER).

Version v1.3.0

January 2014

Change 1: Fixed critical error where USB drive is ejected but not removed.

Change 2: Added SHORTDISPLAY option to PHYSICALRAM, VIRTUALMEMORY, PAGEFILE and FIXEDDISK items.

Change 3: Added ENVVAR item.

Change 4: Added DEFAULTPRINTER item.

Change 5: Added SHORTDISPLAY option to the registry monitor.

Change 6: Added LOGONSESSION item.

Change 7: Moved network adapter filter to the item configuration and is much more flexible with include and exclude options.

Version v1.4.0

February 2014

Change 1: Added ALLIPADDRESS. This is a stand alone item to display all active IP addresses without depending on the NETWORKADAPTER item.

Change 2: Fixed critical error where a network adapter has more than one ip address sometimes causes an access violation.

Change 3: Improved response for changing network adapters, fixed disks and screen resolution.

Change 4: Added hidden items. Set active to 2. Right click and select Show Hidden Items. See above.

Change 5: Added OEMPRODUCT item. See above.

Change 6: Added SUBNETMASK item.

Change 7: Added csv option to items. See above.

Change 8: Fixed disk filter got lost during last version.

Change 9: Apparently fixed an obscure wmi bug.

Change 10: Fixed CPUTEMP charts.

Change 11: Added SERVICESTATE.

Change 12: Added TEXT.

Change 13: Added hook to Core Temp temperature reader.

Change 14: Added custom WMI item. See above for details. There's some examples in the ini file. These may duplicate existing items. I'll leave the old items in for this release and if all goes well they'll be removed next release.

Change 15: Adjusted screen saver display.

Version v1.5.0

March 2014

Change 1: Fixed wmi line chart resetting when switching between hidden and not hidden.

Change 2: Added volume to FIXEDDISK.

Change 3: There was a test item called PROCESSMEM which was hard wired to DesktopInfo.exe. I've modified this so you can point it to any process name. The result will be the sum of all processes by that name. 'ws' is Working Set, 'pf' is Page File Usage or Commit Size.

Change 4: Added 'column1width' to global options.

Change 5: Added format option to WMI item.

Change 6: Added centerv and centerh to options.

Change 7: Removed following items as they can now be done with WMI: SERVICESTATE, AUDIOCONTROLLER, SERIALNUMBER, WORKGROUP. The ini file has been updated as appropriate.

Change 8: I don't know what I was thinking with LOGONSESSION.

Change 9: Added a global exception handler so maybe we won't see any more of those cascading error messages.

Change 10: Added option to adjust background transparency.

Change 11: Cleaned up the display code and fixed a few little annoyances.

Change 12: Fixed IEVERSION to include svcversion updates.

Version v1.5.1

April 2014

Change 1: Fixed charts updating every second even though data is not updated.

Change 2: Fixed regression bug with COMMENT item.

Change 3: Fixed refresh ugliness and missing top lines.

Version v1.6.0

August 2014

Change 1: Major refactor of the internal procedure calls.

Change 2: Reduced the display flickering some more.

Version v1.7.0

August 2018

(Four years, that's not so bad is it?)

There's been a regular stream of inquiries over the years about Desktop Info to warrant looking into resurrecting it. The last official release was v1.5.1 in April 2014. I made available an unofficial version 1.6.0 in November 2017 which was a compile I did in August 2014 and never released. I was amazed to see it working on Windows 10. However some people are having issues so I'm going to make an attempt at resolving those and see where that takes us.

Change 1: Signing with digital code certificate. You can confirm this by right clicking the DesktopInfo.exe, select Properties and the Digital Signatures tab.

Change 2: Removed secondary form from Windows task switcher.

Change 3: Removed MSN status option.

Change 4: Now correctly reads unicode language files. The language files must be UTF-8 encoded.

Change 5: The language files are now collected in the "language" sub-directory. Make sure you specify the sub-directory in the desktopinfo.ini file entry. Send me your language files to be included in future releases.

Change 6: Doubled the size of item text so you can have longer comments.

Change 7: FIXEDDISK now correctly reads *only* local fixed and removable drives, not remote, network, mapped or optical drives. No longer constantly polls drives when the FIXEDDISK item is not active.

Change 8: UPTIME is no longer capped at 49 days.

Change 9: Added ShortDisplay to UPTIME.

Desktop Info - Manual

Change 10: Added UTCTIME. This seems to take a while to run so maybe set a higher interval.

Change 11: Added full width underline style.

Change 12: I made a mistake with the TEXT item. The last field should be 'display' and not 'key'. You should change your ini file to match.

Change 13: Added user variables. See the section above for details.

Change 14: Added csvdatatype option to items. Data logging no longer outputs the data in it's display form by default. The default value of 0 outputs the data in it's raw format. A value of 1 will output the data in the format it is displayed. See data logging section above for more information.

Version v1.8.0

September 2018

This version represents another not insignificant refactor which allows us to do a whole bunch of stuff we couldn't do previously. You should review the readme.txt, there's lots of new information there. There may be things I've broken that I haven't discovered yet. Your current ini file will need modifications to bring it up to speed but it might be simpler to just start again with the new one. Let me know if something breaks that hasn't been mentioned here.

Change 1: The first upshot of the big refactor is the addition of the 'display' template property to every item. This allows you to control how the data is displayed. See the items section above for details. This also means the 'shortdisplay' property is gone; you set your own display.

Change 2: The second upshot of the big refactor is that many items now offer multiple values for display. You decide which values you want by using replaceable parameters in the display template. Not all items have fully implemented this yet, more to come in future releases.

Change 3: The third upshot of the big refactor is a whole new formatting system included with the display template. You now have much more control over numbers, dates, times, bytes, speeds and booleans. See the Formatting sections above for much information.

Change 4: The custom WMI item has also undergone a major renovation and is much more useful now. The 'property' and 'format' options are gone and replaced with the 'display' template option. As with the other items, the WMI display template option allows you to specify the complete text you want to display including any and all returned wmi property values and formatting. See the Custom WMI section for details.

Change 5: The WMI item can now display multiple rows. If your wmi query returns more than one row, additional rows will be added to the display.

Change 6: Added HTTPGET item. This item will make a simple HTTP GET request to the given source url and display whatever it returns. Useful for displaying your public ip for example. See the Http Get section for more information.

Change 7: Removed SCREEN and OSVERSION items. They are returning innaccurate information so we'll use the WMI call instead. See the ini file for the replacements.

Change 8: Removed BIOS, OEMPRODUCT, MOTHERBOARD, PRINTER, PRINTERSTATUS, DEFAULTPRINTER, UTCTIME items. These are all wrappers for WMI calls so now we'll just use WMI directly with some nice formatting. See the ini file for the replacements.

Change 9: Fixed problem reading registry key with a comma in the name. A comma in a registry key should be preceded by a backslash. See above for more information.

Change 10: Added FILEEXIST item. Displays whether the given file exists. See the section above for details and the ini file for an example.

Change 11: Fixed column 1 auto width calculation.

Change 12: Added LOADTIME which is the time of day that Desktop Info was executed.

Change 13: Changed process priority from 'idle' to 'below normal'. I don't expect this to have any impact other than the display refreshes are a little smoother.

Change 14: Speaking of display refreshes, I've hopefully streamlined it some more to reduce unnecessary redraws.

Change 15: If an item color is not defined it will take on the color of the previous item. In this way you set the color of the first item in a block and all successive items will be the same.

Change 16: ALLIPADDRESS now puts one entry per line so if you have multiple IP addresses, there will be multiple lines. Might have found the bug where it crashes when a network interface goes away.

Change 17: Added FILECONTENTS1 and FILECONTENTS2 for displaying the contents of a text file.

Desktop Info - Manual

Change 18: Removed SUBNETMASK and MACADDRESS sub-items and added them as a second and fourth value (%2 and %4) to IPADDRESS.

Change 19: Added prefix as a third value to IPADDRESS.

Change 20: Added subnet mask and prefix as second and third values to ALLIPADDRESS.

Change 21: Now correctly resets it's position when the desktop size changes (maybe).

Change 22: Added 'noresults' option to the language files which is displayed when a query returns no rows, eg WMI.

Change 23: More improvement in language support. All 'text' has been moved out of the desktopinfo.ini file and into the language files. Each key in the language file is either an item id or a language id (lid) in the desktopinfo.ini file. See the section above for more information. Consider the 'text' option in the desktopinfo.ini file deprecated.

Version v1.9.0

October 2018

My primary objective in this release is to rework the raw data infrastructure so that I can store and move all the collected data and have better control over what data is displayed, charted and logged to csv. For now chart data remains pre-selected and the WMI item charts the first two numbers returned. In the next release I will open this up so you can select which values you want to chart. I'm also adding lots of information to the readme file.

Change 1: Reworked the internal data infrastructure for better control.

Change 2: All items now have the potential to be multi line.

Change 3: Implemented line breaks in the display template. Place a pipe symbol, '|' to indicate where you want a line break.

Change 4: Added NETWORKINTERFACES item. This is the alternative to the NETWORKADAPTER item and it's sub-items. It has much more information and display control. Read the item reference for more info. NETWORKADAPTER is deprecated.

Change 5: Reworked the internal item options infrastructure to make it much simpler to add new options to items and pass them on to the procedures.

Change 6: Added 'multirow' option to ALLIPADDRESS, DNSSERVER. This gives you the option of displaying the data over multiple rows instead of all on a single row.

Change 7: Much work on the csv logging. It is now logging all collected data, raw or formatted. Any items that return multiple rows will have all rows written.

Change 8: Added the item reference to the readme file. It should be complete by now. Let me know if something is missing.

Change 9: Removed 'offset' option in the global section of the ini. Added it as an option to the CPUUSAGE, NETWORKADAPTER and FIXEDDISK sub-items.

Change 10: Added DATADUMP item. When this fires it writes a csv file for every active item.

Change 11: Completed implementation of language 'noresults' option.

Change 12: Added the REG item which displays a value from the registry.

Change 13: Added language file change notifier.

Change 14: Added cpu and page faults to PROCESSMEM. Maybe it should be PROCESSMON?

Change 15: Updated the debug logging for all items and display data.

Change 16: Added 'logo' png graphic header to the language file. I know you don't like my silly logo. You put your own logo in.

Change 17: Font config has been moved out of the desktopinfo.ini file to the language files. This is so you can set a font for a specific language.

Change 18: Added font-charset to font config in language files. I'm still learning how to do languages so this might take a couple of iterations.

Change 19: Added language config to the context menu.

Version v1.9.1

October 2018

Change 1: Resolved some Unicode issues. Debug log is now unicode.

Desktop Info - Manual

Change 2: Fixed issue with not resetting it's position after display size change.

Version v1.9.2

October 2018

Change 1: Fixed some fatal issues with the FILE/REGISTRY/EVENT monitor items.

Change 2: Fixed formcolor bug.

Version v1.10.0

November 2018

Important changes from v1.8.0 and v1.9.0: In those versions I started moving config options out of the main ini file and into separate language files. I'm not happy with the way that's working out so I've returned to making the desktopinfo.ini file the primary configuration file and the language files will simply override that if required. In most cases you will not need a language file at all. My apologies if I made work for you.

In this release we continue to reap the benefits of the raw data infrastructure refactor done in v1.9.0 with major improvements in both the charting and threshold functions. I've also been working hard on language support so we can now display non Latin based languages.

Change 1: Revamped the thresholds so we can have up to 3 thresholds on each item using any returned value *plus* we can have negative thresholds *plus* we can have thresholds on WMI items. See the [Thresholds](#) section for more info.

Change 2: Revamped the charts so we can chart any numerical value from any item including WMI items, bar or line, linear or logarithmic, colors, thresholds etc. See the [Charts](#) section for details.

Change 3: Added [REGEXIST](#) item.

Change 4: ALLIPADDRESS has been extracted from it's dependency on NETWORKADAPTER, it's list and it's 'activeonly' and 'filter' options. It will now work independently. NETWORKADAPTER is deprecated.

Change 5: I've taken the 'hidden' page of items and turned it into a whole series of pages. Previously, "active:2" was the special 'hidden' page. This is now simply page 2 out of 9 pages. See the [Pages](#) section for more info.

Change 6: The readme.txt file was getting too big. Time for a proper manual.

Change 7: Added [FILE2TEXT](#) item.

Change 8: [OEMINFO](#) now checks both 64 bit and 32 bit branches of the registry tree.

Change 9: Added [log-level option](#).

Change 10: Splitting the config between two files was a bad idea so I thought it out again and the desktopinfo.ini file returns to being the primary configuration file. Everything is configured within that file. If you wish to override any options or text without altering the main configuration, create a language file. See the [Language File](#) section for more information.

Change 11: Much work on unicode support means we can now create and display non Latin languages such as Croatian, Chinese, Japanese etc via the language files. There is still work to be done to make the main ini file fully unicode so for now you must use the language files for multi-byte text.

Change 12: Added right-to-left support for Arabic languages. It's not complete but it is usable. If you find any errors in any of my sample language files please let me know.

Change 13: More work reducing the flicker/shimmer.

Change 14: The package no longer contains the desktopinfo.ini file, the file is renamed to desktopinfo-sample.ini. This is in order to prevent accidentally overwriting your finely tuned custom config file when the new version arrives. When Desktop Info runs, if there is no desktopinfo.ini file, it will copy the sample over and start using it. This makes it transparent for new users. Similarly, the packaged language files are in the "sample-languages" directory. If you decide to use one of the sample language files, you should copy it to a new location first to avoid it being overwritten by the next new version.

Change 15: I've greatly pared down the sample ini file because, well, it's just a sample and there's too many options and items to include everything. Good thing there's a proper manual now to see what's available.

Change 16: Added mtu option to [NETPACKETS](#) and [NETPACKETSRATE](#).

Change 17: Added basic [Windows Event Logs](#) logging.

Change 18: Fixed "InetNtpw not found ws2_32.dll" bug running under Windows XP.

Desktop Info - Manual

Change 19: Skip some [NETWORKINTERFACES](#) values not available under Windows XP.

Change 20: Added [logo-align](#) to options.

Change 21: Added queue length to [CPU](#) item.

Version v1.10.1

November 2018

Change 1: Fixed bug with logo not updating after ini change.

Change 2: Fixed bug with not aligning correctly when the 'bottom' option is used.

Change 3: Fixed contextmenu option not working.

Change 4: More work converting internal routines to Unicode.

Change 5: I believe I've resolved the stability issues. If you leave the "log-level=error" option on in the ini, you'll see when an error occurs and hopefully the reason.

Version v1.10.2

November 2018

Change 1: Fixed bug with FIXEDDISK and NETWORKADAPTER spinning out of control under Chinese Windows.

Change 2: Continue the Unicode work.

Change 3: Added reference English language file.

Change 4: Fixed handle leak in ProcessMem, TopProcess and registry routines.

Change 5: Fixed bug in bar chart when only one data point is available.

Version v1.11.0

December 2018

There are two important architectural changes in this release: I've moved the data collection code into a thread to allow the gui to be more responsive. This has the side effect of raising the memory footprint slightly. Secondly, I've added a background process to collect 64 bit metrics. This is controlled from the ini file. Plus a bunch of new stuff to explore.

Change 1: Fixed a couple of pointer leaks in options loading and registry/event/file watchers.

Change 2: Changed load model for a couple of DLLs which were causing issues.

Change 3: Added counter option to [TOPPROCESSMEM](#).

Change 4: Added private usage to [PROCESSMEM](#).

Change 5: Introducing the [DesktopInfo64](#) tool. This tool runs silently and allows me to retrieve 64 bit counters and other metrics not available to 32 bit applications.

Change 6: Fixed startup failure on Windows 2000.

Change 7: Added hexadecimal [number format](#) option.

Change 8: Updated reference English language file. If you are a translation contributor please check out your language file to ensure it's accurate and up to date.

Change 9: Added the [BIT](#) display format.

Change 10: Added host option to the [REG](#) and [REGEXIST](#) items.

Change 11: Added support fields to [OEMINFO](#).

Change 12: Added the [monitor-mode](#) option. This gives the option of using Primary Monitor mode or Virtual Screen mode. Previously DTI was using the Primary Monitor mode and this remains the default mode.

Change 13: Added [navigation buttons](#). This really highlighted how unresponsive the gui was so...

Change 14: Added data collection thread to make the gui more responsive. Because of this separation, I've returned the application priority class to idle.

Change 15: More Unicode work in the ini file.

Change 16: Moved the [options] text strings into their own [\[text\] section](#). You will need to move your 'noresults' option to this new section in order for it to be recognised. See the sample ini.

Change 17: Added 'error' [text string](#).

Desktop Info - Manual

Change 18: Added [SESSIONID](#) and [RDSSESSIONID](#).

Change 19: Changed the TOPPROCESSCPU algorithm to be more accurate.

Change 20: Fixed bug where DTI was responding to Alt+F4 meant for another application.

Change 21: Added [text-editor](#) option.

Version v1.12.0

January 2019

This release has no major architectural changes, yay! There's some minor tweaks to smooth out the navigation buttons and paging, some new tools and bug fixes and a change to how monitor-mode works. If you use the monitor-mode option you'll need to check out the changes.

Change 1: Fixed [REGISTRY](#) so it can read "read only" keys.

Change 2: Now responds to display changes and hopefully also remote desktop display changes.

Change 3: Added the [IF](#) item. See [Conditional Visibility](#) for a full description.

Change 4: Added the '[hidden](#)' common option to items. This allows you to retrieve data without necessarily displaying it. The SET and IF items are hidden by default. Useful for building composite items and doing conditional visibility.

Change 5: Fixed incorrect form height when using navigation buttons in "form" mode (/f).

Change 6: Fixed flickery navigation buttons.

Change 7: Added [text-align](#) option to [options].

Change 8: I've changed monitor-mode so that in virtual screen mode, the origin (top=0, left=0) is the top left of the virtual screen world rather than the top left of the primary monitor. You should review the [size and position](#) description to understand how it works now.

Change 9: Added [monitor-num](#) so we can place Desktop Info on a specific monitor. This option is not used when monitor-mode=1.

Version v1.13.0

February 2019

Change 1: Added [line-spacing](#) to [options]. This controls the number of pixels between rows so you can squeeze it closer together or spread it out.

Change 2: Fixed csv bug duplicating header cells.

Change 3: Added maxrows and %rowcount% to the [WMI](#) item. In addition to limiting the output of the WMI display, it also provides a way to display just the row count rather than all the rows. Keep in mind this still requires a full parse of the result set so a large WMI query will take some time to process even if just getting a row count.

Change 4: Added the [text-offset](#) common option to all items. The *offset* option of the [NETWORKADAPTER](#) and [FIXEDDISK](#) child items has been removed and they now have a default text-offset of 10 pixels. You can set this to any number to adjust the offset or 0 if you don't want an offset.

Change 5: Fixed the offset bug in right aligned and right-to-left text.

Change 6: Added rgb option to [item colours](#).

Change 7: In WMI queries, CIM_DATETIME fields are automatically split into the date and time components so you can format them properly in the display template. See [WMI Date/Time Formatting](#).

Change 8: Added a sample 'advanced' ini file with multiple pages. You can either rename it to the main ini file, desktopinfo.ini or add it as a command line parameter: desktopinfo.exe /ini=desktopinfo-advanced.ini.

Change 9: Fixed header bug in [VIRTUALMEMORY](#) csv and added free and percent free to results.

Change 10: Fixed header bug [PAGEFILE](#) csv and added free and percent free to results.

Change 11: Added free and percent free to [PHYSICALRAM](#).

Change 12: If there is one option on the command line with no '/' prefix, it is assumed to be an ini file. This way you can drag an ini file onto the exe file to start Desktop Info using that ini file.

Change 13: Added 'pages read from disk per second' and 'pages written to disk per second' to [PAGEFAULTS](#).

Change 14: Allow the [bar charts](#) to be on their own row so that they are full width. Add two pipe symbols to the end of the display template.

Desktop Info - Manual

Change 15: Added the [bar-colors](#) option to customise the bar chart colors.

Version v1.13.1

February 2019

Change 1: Fixed display bug in WMI number transforms.

Version v.1.14.0

March 2019

Change 1: Added the 'key' option to [FILE2TEXT](#) to retrieve a single value from a file.

Change 2: Allow absolute path in [language file](#) option.

Change 3: Added user variables to the [WMI](#) query.

Change 4: Fixed crash due to null property in WMI results.

Version v.1.15.0

April 2019

Change 1: Added handle count to [PROCESSMEM](#).

Change 2: Added Unix text file (LF) support to [FILE2TEXT](#).

Change 3: Added Unix text file (LF) support and multibyte support to [FILECONTENTS1](#) and [FILECONTENTS2](#).

Change 4: Finally got the main ini file multi-byte compatible. Now you can set the *text* and *display* options in the main ini file using any language provided you save the ini file as UTF-8. If your language is right-to-left, set the *font-rtl* option in the [options] section.

Change 5: Added [CONTROL](#) item. This provides button and hyperlink controls that will open the given resource (web page, document, folder, application etc.). The controls are not yet multi-byte compatible. There are examples in the desktopinfo-advanced.ini config file.

Change 6: Various stability fixes.

Change 7: Search for the supplied ini file in the application directory.

Change 8: Fixed crash with threads writing to the log file.

Change 9: Debug log level shows all entries.

Change 10: Fixed bug in [ALLIPADDRESS](#) with multi port NICs showing the same IP for both ports.

Change 11: Some changes to [NETWORKINTERFACES](#). Where an interface has multiple IP addresses, return values %8 through %13 will show the first ipv4 and ipv6 addresses (it used to show the last one). The return values %15 through %22 (dns server etc) will all show a space delimited list of IP addresses.

Change 12: Added new options and return values to NETWORKINTERFACES to handle [network interface cards with multiple ports](#) and hence multiple IP addresses.

Change 13: Enhanced the [EVENTLOG](#) item to return more useful information. This used to run in a 'watch' thread, I don't really know why, but it's now a regular item.

Change 14: Added [SNAPSHOT](#) item to create a snapshot image of the current DTI display.

Change 15: During refresh, only reset the form position in normal display mode, never in "/f" mode.

Change 16: Added fallback in case no items were read from the config file.

Change 17: Added search for file in application directory to [FILE2TEXT](#), [FILECONTENTS1](#) and [FILECONTENTS2](#).

Change 18: Added *read hit ratio* and *read miss ratio* return values to [PAGEFAULTS](#).

Change 19: Fixed some colour issues in "/f" mode.

Version v.1.15.1

May 2019

Change 1: Fixed user variables to be multi-byte compatible.

Version v1.16.0

June 2019

Following various fixes in this and the last couple of releases, I find the general stability of Desktop Info to be quite good. However it's a complex program with lots of moving parts and I may be missing something. If you are still experiencing random crashes or DTI mysteriously vanishing, I would be interested to work with you to get to the root cause of the problem. Drop me a note on the forums.

Change 1: Added [expressions](#) and [user defined functions](#) to display templates.

Change 2: Fixed [best fit number format](#) where 1000M should be 1G etc.

Change 3: Added [null-result text string](#).

Change 4: Fixed WMI property value where very large number was being set to null.

Change 5: Fixed WMI where a null property display is processed incorrectly.

Change 6: Added hide-no-result option to item [common options](#).

Change 7: Added [CMD](#) item to execute external tools such as cmd.exe or powershell.exe etc.

Change 8: Added font-size to item [common options](#). Now you can set the font size for each item. The font-size in the [\[options\] section](#) remains the global font size but now each item can override it. Because the code controlling the font sizes and general layout has changed, the default value for the [line-spacing](#) option is now 1. No font-face option yet, maybe later.

Change 9: Added *hide-key* option to [FILE2TEXT](#) item.

Change 10: Added user variables to the key option in [FILE2TEXT](#).

Version v1.17.0

July 2019

Please take note of deprecated items as they will be going away. In the next release, v1.18, BATTERY and CPUTEMP will be removed. You should replace these with standard WMI items. See the [Useful WMI Queries](#) chapter for examples. Also in v1.18, NETWORKADAPTER will be removed. You should replace this with NETWORKINTERFACES. I will also be building a new item LOGICALDRIVES to replace the FIXEDDISK and DISKIO items and so they will also be deprecated and eventually removed.

Change 1: Fixed font sizing on controls.

Change 2: Added *font-face* to item [common options](#).

Change 3: Fixed control alignment.

Change 4: Added [\[screen-saver\]](#) section to the ini file. This has all the same options as the [\[options\]](#) section but is used in screen saver mode. This means the screen saver is somewhat more flexible now.

Change 5: Added user variables to [FILE](#), [FILE2TEXT](#), [FILECONTENTS1](#), [FILECONTENTS2](#), [FILEEXIST](#).

Change 6: Fixed a bug in FILE2TEXT that probably stopped it working altogether.

Change 7: Fixed a fatal crash on startup when the Processor performance counter is missing. Thanks to ntech2 for help tracking this one down.

Change 8: Finally got around to testing the [ipv6list](#) option in [NETWORKINTERFACES](#) and confirmed it is returning a full list of all ipv6 addresses.

Change 9: Added *text-align* to item [common options](#) so you can override the global *text-align* option on a per item basis.

Change 10: Added *initial-page* option to the [\[options\]](#) section. Particularly useful in the [\[screen-saver\]](#) section.

Change 11: I've built a new unicode hyperlink [control](#) and also updated the button control for unicode.

Change 12: Added *include-all* option to [FIXEDDISK](#). This gives you the option to include the floppy drives a: and b: in the fixed drive list. Default is off.

Change 13: Added option to specify [item font-size](#) as percent of the default font size set in the [\[options\]](#) or [\[screen-saver\]](#) sections.

Version v1.18.0

July 2019

Change 1: Fixed crash in collector logging during config refresh.

Desktop Info - Manual

Change 2: Refactor of form creation code fixes a number of issues with form display in the three modes: desktop/window/screen saver.

Change 3: Fixed obscured column 1 text on button control.

Change 4: Removed deprecated BATTERY item. See [Useful WMI Queries](#) for a replacement.

Change 5: Removed deprecated CPUTEMP item. See [Useful WMI Queries](#) for a replacement.

Change 6: Removed deprecated NETWORKADAPTER item. Use [NETWORKINTERFACES](#) instead.

Change 7: The [SET](#) item is now global. That is to say it will be evaluated every cycle regardless of the active and interval options and what page is being displayed. You can view the result of the SET evaluation by setting hidden:0 and active to the page upon which you want to display it.

Change 8: The [SET](#) item is now evaluated during configuration load so that user variables may be used to store color values.

Change 9: Added total up time in seconds to [UPTIME](#).

Change 10: Moved alarm checking into the collector. It also works out which display line contains the alarming metric.

Change 11: The threshold color change is now restricted to the display line on which the metric is appears and only column 2 text, not column 1 title text.

Change 12: Added new item [LOGICALDRIVES](#) which will replace [FIXEDDISK](#) and DISKIO.

Change 13: Added option to display the [navigation buttons](#) at the top of the display, below the logo if visible.

Change 14: Restricting config reload to DBT_DEVICEARRIVAL and DBT_DEVICEREMOVECOMPLETE for DBT_DEVTYP_VOLUME and only when FIXEDDISK is active. This completely eliminates the redraws when a drive is plugged in or removed when using the new LOGICALDRIVES. Otherwise only a DBT_CONFIGCHANGED will trigger a config reload.

Change 15: Added Mod function to [expressions](#). Modulus returns the remainder after dividing the first argument by the second argument.

Change 16: Added unix time to [DATETIME](#) and [BOOTTIME](#).

Change 17: Added expression parser to user variables so you can do some math when setting a variable value.

Change 18: Added expression parser to [IF](#) item so you can do some math when comparing values.

Change 19: Removed the single instance restriction and the Show Desktop option on the context menu.

Version v1.19.0

September 2019

Change 1: [CONTROL](#) item now responds to the *style* common option. An unexpected side effect is the control can be put in the left column by using the 'w' style option.

Change 2: Color variables are available for the [bar-colors](#) option.

Change 3: Fixed line spacing issues.

Change 4: Fixed bug in [LOGICALDRIVES](#) *count* option. This was getting confused with another count option so I've changed it to *max* to avoid conflict.

Change 5: Fixed Clear Type issue with hyperlink controls.

Change 6: Added user variable processing to the COMMENT item and most other items that have a *text* option for the left column.

Change 7: Bar chart [positioning](#) has changed. The chart used to be placed at the end of the item display. Now it can be placed anywhere within the item display by placing the %chart% marker in the *display* template option. This also fixes the charting problem introduced with LOGICALDRIVES in v1.18. Your current config may need to be modified.

Version v2.0.0

October 2019

I've decided to take a leaf out of Linus' book and move to version 2. Not because of any major new developments but because we've been on version 1 for long enough and it's sufficiently different from the version 1 of nine years ago.

Change 1: Environment variables in the *uri* option of [CONTROL](#) are now resolved prior to execution.

Change 2: A new *args* option has been added to [CONTROL](#) for additional command line arguments.

Change 3: Fixed bug in [LOGICALDRIVES](#) where the *set* variable contained an extraneous pipe at the end.

Desktop Info - Manual

Change 4: Added thread count to [PROCESSMEM](#).

Change 5: Minor crash fix in mapping code.

Version v2.0.1

October 2019

Change 1: Fixed high cpu issue in 64 bit tool.

Change 2: Threshold colors now correctly support rgb format.

Change 3: Fixed large value threshold bug.

Change 4: Added support for escaped ampersand, “\&”, in addition to double ampersand, “&&”.

Version v2.1.0

December 2019

There are significant architectural changes in this release in order to get logical row titles and data in the left column and also to get the CMD output displaying correctly so pay close attention, there's an excellent chance I've broken something and your configuration file will most likely need to be changed. You should test well before putting it into production and let me know what you find. Also of special note, the deprecated FIXEDDISK item has been removed.

Change 1: Refactored [CPUUSAGE](#) to a regular multi-row item. Replaced the *count* option with the *maxrows* option, changed the return values. If you're using the bar chart with this item you will need to change it from property 1 to property 2.

Change 2: [Banner graphic](#) now supports many image formats.

Change 3: In the banner graphic, I believe I have resolved the issue of artefacts appearing around transparent png images.

Change 4: [Items](#) can now be split over multiple lines in the config file by ending the line with a backslash character as per Python, Bash etc to indicate it is continued on the next line.

Change 5: Added *drive-size* option to [LOGICALDRIVES](#).

Change 6: Refactored the [CMD](#) item so it reads STDOUT directly. You no longer need to redirect to a temporary text file first. Return value %4 is now the default display template which will display the cmd output. Added the *wide* option so the output can be displayed over both columns. See the CMD example on page 6 of the desktopinfo-advanced.ini sample file for a good PowerShell example.

Change 7: Added total threads to [PROCESSCOUNT](#).

Change 8: Added the *row-text* option to provide a way to create a logical row title in the left column. This option can include return values to display row data in the same way as the display template. For example to display the drive name in LOGICALDRIVES or the cpu id in CPUUSAGE. See the [discussion here](#).

Change 9: The *multirow* option in the [ALLIPADDRESS](#) and [DNSSERVER](#) items now defaults to 1.

Change 10: FILECONTENTS1 and FILECONTENTS2 are replaced with a single [FILECONTENTS](#) with the *wide* option to control whether it's displayed in the right column or over both columns.

Change 11: Having achieved the final hurdle of getting row titles in the left column, the deprecated item FIXEDDISK has been removed. You should use [LOGICALDRIVES](#) instead. Along with the refactored CPUUSAGE, this is the last of the weird parent/child items.

Change 12: Added *maxrows* to [TOPPROCESSCPU](#) and [TOPPROCESSPF](#) so you can have more than one top process.

Change 13: Might have fixed the bug where occasionally a control would hang around after the page has changed.

Change 14: Updated the desktopinfo-advanced.ini with some new pages.

Version v2.2.0

December 2019

Change 1: Fixed broken [FILE2TEXT](#).

Change 2: Fixed broken [TEXT](#).

Change 3: [CMD](#) is now Unicode aware. The target command executable must also be Unicode aware for this to be effective. I am yet to figure out exactly how to initiate a Unicode Powershell process so I don't know if this works for Powershell or not. Perhaps if you work it out you can let me know.

Desktop Info - Manual

Change 4: Fixed the confused *font-rtl* and *text-align* options. They are now separated and work independently. Both may be added to the [\[options\]](#) section as global defaults. Both may be added to [individual items](#) to override the defaults.

Change 5: I've begun adding some [key words](#) to the items section to make it easy to set item default values and try to reduce some of the complexity. For example, by setting *page=1*, you do not need to specify the active option for each item on that page.

Change 6: The old method of an item setting the current default colour is replaced with the new COLOR [key word](#) for setting the default colour. If you've been relying on this method to set default colours, you'll notice some colour oddness. Just set the new COLOR key word at an appropriate spot in the items.

Change 7: I finally got around to making the context menu Unicode and configurable. You can adjust the text of the menu options (Refresh, Quit etc) using the [\[text\] section](#) in either the main ini file or the language ini file. I also added the ability to remove an option from the menu by setting it's text to blank. Updated language files. Let me know if you think the translations need improvement.

Change 8: The page titles in the context menu are also Unicode aware. You can set the page titles using the [\[pages\] section](#) in either the main ini file or the language ini file. You can also set page titles on the fly using the new PAGE-TITLE [key word](#). Updated language files.

Change 9: User variable names are no longer case sensitive.

Version v2.3.0

January 2020

Change 1: Refactored the [thresholds](#) to support return values, user variables and expressions.

Change 2: Increased the number of thresholds to nine.

Change 3: Added the *filter* option to [TOPPROCESSCPU](#) and [TOPPROCESSPF](#).

Change 4: Added *maxrows* and *filter* options to [TOPPROCESSMEM](#). In 64 bit mode *maxrows* is capped at 10.

Change 5: The *filter* options are now Unicode.

Change 6: Improved error trapping and recovery in mapping file view that occasionally causes it to fail when retrieving commit size and thread count from desktopinfo64.

Change 7: Some speed optimisations.

Change 8: Enabled RGB colors in line chart series.

Change 9: Added the [background-interval](#) option to all items. This allows you to collect data in the background for any item that is not on the currently active page. It allows you to keep recent data available, keep line charts ticking over and write csv data files for any item on any page.

Change 10: Added a check to ensure LOGICALDRIVES performance samples are at least one second apart. This fixes the bug where a group of such items would fail to display performance data for the second and successive items.

Change 11: Removed TSSESSIONS, SESSIONID and RDSSESSIONID items and replaced them with [RDS](#) and [RDSSESSIONS](#) items.

Change 12: Added [auto-home](#) option to [options]. This returns to the home page after the specified seconds of inactivity.

Change 13: Added unix time to [LOADTIME](#).

Change 14: Added short and long date and time return values to [DATETIME](#), [BOOTTIME](#) and [LOADTIME](#). The default display template for the three is "*display:%9 %10*" which is the short date and time.

Version v2.3.1

January 2020

This is a bug fix release targeting specific user bug reports.

Change 1: Fixed bug where the left column text would not display if there are no results.

Change 2: Fixed bug where ini reload is not resetting the default page number.

Change 3: Added the *code-page* option to the [CMD](#) item so that tools returning Unicode text such as Powershell can be displayed correctly. If *code-page* is specified, the text will be decoded using that code page. If *code-page* is not specified, the text is tested for UTF-8 and handled accordingly. Otherwise the text is assumed to be ASCII text.

Change 4: Fixed a bug in debug [logging](#).

Version v2.4.0

March 2020

Change 1: Added the common item option [text-color](#) which allows us to specify a color for the left column text.

Change 2: Added the key word [TEXT-COLOR](#) which allows us to specify a new default color for the left column text for all following items. This is distinct from the item option *text-color* above. See the discussion at [Item Colors](#) for an explanation of how it all works. If you do nothing, your current configuration will continue to work as it has.

Change 3: Added the *wide* option to [CONTROL](#) instead of using the wide font style to be consistent with other items.

Change 4: Added the *left* and *width* options to CONTROL.

Change 5: The [auto-home timer](#) is reset after a config change and reload.

Change 6: Removed redundant volume changed code since the last code that used it is gone.

Change 7: Added [CONTROL2](#) item. It is identical to CONTROL except it doesn't add a new display row, it is added to the display row of the previous CONTROL item. I'm still working on RTL for controls. I hope to have that sorted out in the next release.

Version v2.4.1

March 2020

There was a fatal flaw in the new CONTROL code which threw an exception and sent the application into a tail spin. This has been resolved along with a couple of other minor code cleanups.

Version v2.4.2

March 2020

Change 1: Fixed bug where alarms were not changing color.

Change 2: Fixed bug where threshold would fail if property was the last one in the list.

Version v2.5.0

April 2020

What started out as a couple of quick fixes to deal with display issues has turned into a major refactor of the display routines. Fixes that work on one machine don't work on another so it's taken a bit of time to figure out what works in all cases.

Change 1: Button and hyperlink controls modified to reduce redraws.

Change 2: Major refactor of the display routines which eliminates the blinking and shimmering that some people are experiencing. For those in the know, the display now uses a masked buffer bitmap (ala Buster Keaton), similar to double buffering but more sophisticated. A lot of the logic has changed, often in subtle ways, so watch out for introduced bugs.

Change 3: Refactored the bar chart drawing routine to eliminate the bouncing effect.

Change 4: Adjusted the thread priorities. The main thread is now below normal priority and the collector thread is low priority.

Change 5: Do not load empty language string to avoid empty displays when using a language file. When you enable a language file, all the comments would immediately stop displaying because the comment item in the language file is empty. Now, if a language item is empty it will keep the item text from the main ini file.

Change 6: Fixed [RDS](#) item to correctly display ipv4 address. See the notes in the item reference for more information on how RDS displays the client ip address.

Change 7: More work on RTL and right alignment issues.

Change 8: If you've dragged DTI to a new location, changing the page won't send it back to it's starting position.

Change 9: Navigation buttons modified to reduce redraws.

Change 10: The navigation buttons no longer grow (highlight) when you hover the mouse. However, I've made them a slightly bigger target and the mouse cursor still changes when you hover over them so you should be ok.

Change 11: Added a minimum width constraint on the form mode (/f).

Change 12: [HOST](#) item is now multi-byte.

Change 13: HOST item is extended to include dns domains and host names.

Change 14: Added navigation button hover hint language strings to the [text](#) section.

Desktop Info - Manual

Change 15: Fixed bug where switching languages on the fly doesn't always update the context menu.

Change 16: [CONTROL](#) buttons and hyperlinks now support user variables in the *button-color* and *hover-color* options as well as the *display* and *uri* options.

Change 17: The [context menu](#) Refresh option is now Reload because instead of just refreshing the display it does a full ini file reload.

Change 18: Added [PHYSICALDISKS](#) item. This shows all physical disks, size, manufacturer, partitions, logical drives etc.

Change 19: If the *code-page* option is specified in [CMD](#), it will try to remove the leading byte order mark (0xEF 0xBB 0xBF) from the output if it finds it.

Change 20: Added the *trim* option to [CMD](#) to try to trim leading and trailing white space and line feeds from the output. Hopefully this will help cleaning up Powershell output.

Change 21: Added the *sample-scripts* sub-directory which will contain various interesting Powershell scripts and batch files.

Version v2.6.0

May 2020

Change 1: Fixed introduced bug where the display would fail to update if nothing was displayed on the initial refresh.

Change 2: Added remote monitoring option. This provides a mechanism for Desktop Info to retrieve raw data collected by another instance of Desktop Info. See the [Options](#) section to switch on the server and the [DTI](#) item for retrieving remote data. Also, a [detailed discussion here](#).

Change 3: Trimmed *get-windows-updates.ps1* to a single query. It was taking a long time.

Change 4: Changed default button color.

Change 5: Added About box to context menu. It shows the current version and checks the web site to see what the latest version is. This can be hidden by setting it's [text](#) to blank.

Version v2.6.1

May 2020

This release specifically addresses issues with the IF and CONTROL/CONTROL2 items and how they behave and interact.

Change 1: The [IF](#) item will now evaluate every cycle. Therefore the *interval* option is redundant.

Change 2: The relationship between [CONTROL](#) and [CONTROL2](#) is more complete and robust. I discovered a fatal flaw in the config loading code causing pointers to wander off into fairy land.

Change 3: The IF item now correctly sets CONTROL item visibility where appropriate. See the [IF](#) item for an example. Also, because of the strong parent/child relationship between CONTROL and CONTROL2, when you set the visibility of a CONTROL item, the child CONTROL2 items will follow. Also, you can't use an IF item on a CONTROL2 item, it will be applied to the next non-CONTROL2 item.

Change 4: The CONTROL options, *button-color* and *uri* now correctly parse user variables on the fly.

Version v2.7.0

July 2020

Change 1: Added an [Export](#) tool to the context menu. This exports all active, non-hidden items to a file using the selected template file. The menu is populated with files found in the *templates* directory. You can hide it by setting *menu-export* in the [\[text\] section](#) to blank. The resulting file is automatically opened in the default viewer for the file type. From there you can print it or save it to your preferred name and location. Remember that initially not all items have data, you might need to scroll through the pages once before doing an export. It does not export charts.

Change 2: Resolved problem with [hyperlink](#) forcing underline on it's accompanying text. Now the default is no styling on the text and underline on the hyperlink. The style option will be applied to both the text and the hyperlink and additionally, the actual hyperlink will always be underlined.

Change 3: Updated language strings.

Change 4: Found a few more functions that needed to be updated for multi-byte.

Change 5: Fixed bug in WMI date/time field processing that was mangling any WMI query with a date/time field.

Desktop Info - Manual

Change 6: Expanded the options for specifying a chart series. A series can now be specified using the traditional item return value number, a “%” style number, a WMI return value field name, a user variable and an expression. See [Charts](#) for details.

Change 7: Added procedure run time and item id to debug output.

Change 8: Added alternative cpu count return value, %2, to CPUCOUNT. This may or may not return the same or different value.

Version v2.8.0

August 2020

I’ve started refactoring the charts to make them more flexible. I’m not there yet but this version does surface some benefits of the changes. Because there’s changes to underlying chart code, you should check your existing charts to make sure they’re all still as they should be and let me know if you find anything “of interest”. I’m also building a scripted test harness that hopefully will help find regression errors before shipping. I know what you’re thinking and I’ve been suggesting this to our large development team for ages but they’ve only just started thinking about it.

Change 1: Added [ignore-setting-change](#) option. This will suppress some redraws by ignoring Windows WM_SETTINGCHANGE messages triggered by events such as desktop background color changes and remote desktop session connects.

Change 2: Added [control-style](#) option to hyperlink controls.

Change 3: You can now place additional text after the [bar chart](#) %chart% place holder in the item display option. The bar chart width will shrink accordingly.

Change 4: In [negative bar charts](#), the colors are no longer hard coded. The color is now calculated from the data value and the bar-colors option and could be anywhere on the gradient.

Version v2.9.0

October 2020

The main focus of this version is the new improved charts. The old charts are still supported for the time being but will eventually go away so you should change over to the new charts as soon as you can. The [\[pages\] section](#) and the [active item option](#) will eventually be removed in favour of the [PAGE](#) and [PAGE-TITLE](#) keys entered throughout the [\[items\]](#) section. You should update your ini files now to avoid disappointment later on.

Sometimes I miss something in the manual that should have been updated to reflect changes to the application so if you see any out of date information, please let me know so I can bring it up to date.

Change 1: Added the ‘unknown’ [text_string](#) for situations where item results are incomplete such as LOGICAL_DRIVES file system and volume name.

Change 2: Added bar2 and line2 charts. These are more flexible and overcome a lot of the limitations of the old charts. Both can chart every data row of a result set. Bar charts can display two series and line charts can display three series. Remember, multi-row line charts require the *row-id* option. See the new [Charts](#) section for details. The old charts are still available but are deprecated and will eventually go away. See [Legacy Charts](#). For a quick overview on moving from old charts to new charts, see the [Moving To The New Charts](#) section.

Change 3: The sample ini files have been updated to the new charts.

Change 4: If the %chart% placeholder is not present in an item display template, it will be added to the end with a line feed so the chart will occupy it’s own display row after the item data. However, you can place it anywhere in the display template. If you’re using an older ini file this may result in a blank line before the chart. Just remove the extra line feed pipe from the end of the display template.

Change 5: If a chart is displaying a metric that is alarming then the chart frame will change color as well.

Change 6: Extended the item *set* option to allow setting multiple variables from the raw data. See the [Item Set Option](#) section for more details.

Change 7: Added more flexible ways to define values in the [threshold](#) item option such as user variables, expressions, percent style return value numbers and wmi style return value names.

Change 8: Added a last resort attempt to convert the CMD output to a number so it can be charted. That is to say, the [CMD](#) output is expected to be a string for display but it might return a single numeric value that can be charted.

Version v2.10.0

November 2020

Change 1: Added check for byte order marker when reading the ini file.

Change 2: Fixed WMI Uint32 bug.

Change 3: Updated reference for [ENVVAR](#) item.

Change 4: Changed the default *text* value for ENVVAR to “%1”.

Change 5: If [row-text is not defined then text](#) is used. If *text* is not defined then the internal default is used. I’m not sure if this is how it was but it’s how it’s supposed to be.

Change 6: Fixed no results bug in [SERVICEPACK](#) item.

Change 7: Fixed bug in one section of the alarm [threshold](#) parsing preventing use of wmi property names when determining the frame color.

Change 8: I’ve learnt some things about font anti-aliasing and so I’ve removed the *cleartype* option and replaced it with the [font-quality option](#) which gives us an anti-aliasing scale from 0 to 5. The default is 4 which is the best anti-aliased quality short of clear type. If you want to enable clear type, add “*font-quality=5*” to the options.

Change 9: Added the [read-as-csv option](#) to the [CMD](#) item. This allows you to display and chart data from any tool that can output data to stdout in csv format, for example, `nvidia-smi`.

Change 10: Added the [FILECSV](#) item to read in external csv files and display the contents as multi-row data.

Version v2.10.1

January 2021

Change 1: When using the context options to open the ini file and language file, make sure the file parameter is quoted.

Version v2.10.2

March 2021

Change 1: Added environment variables to the [FILE](#) item.

Change 2: Added user variables to [REGISTRY](#) item.

Change 3: Added multi SET processing to WMI items.

Change 4: Fixed series percent bug in bar chart.

Change 5: Added new option [ALIGN](#) at both the global and item level and altered TEXT-ALIGN to override the left column. See [Text Alignment](#) for an explanation.

Change 6: Ensure controls respond to alignment.