Desktop Info
by Glenn Delahoy

Web Site: www.glenn.delahoy.com

Table of Contents

Description

This little application displays real time system information on your desktop. Perfect for quick identification, walk-by monitoring and simple remote monitoring of production servers, test farms or your own personal computer. Great for techs to quickly acquire critical system information of patient machines. Uses little memory and low cpu. There's lots of options. Everything is customisable.

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

Windows Platforms

Desktop Info version 3 is tested on the following Windows platforms:

XP Pro (32 bit) version 2002 service pack 3
Server 2012 R2 (64 bit)
Server 2016 Server
Windows 7 Professional 64 bit
WinPE 10 1809 x86

Windows 10 Home 22H2
Windows 11 Home 22H2

Desktop Info version 3 does not work on Windows 2000. If you really need to use Windows 2000 then Desktop Info version 2 is for you.

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.

Desktop Info Secure Installation

The following was posted on the forum by a user and may be of interest to those needing to maintain a secure environment.

"Hi, I want to report a vulnerability RELATED to Desktop Info, not in it. I'm a professional penetration tester. Due to the lack of an installer, I've seen my customers unzip the program to the root of the C drive. Any new directories created on the root of the C drive have weak permissions which gives "authenticated users" read/write permissions, unlike those placed under "Program Files". This allowed me to escalate privileges from a Domain User to Domain Administrator as well as local system privilege escalation.

Since there was no installer, the program was unzipped to C:\DesktopInfo and a shortcut placed in the Windows startup folder. I simply added CMDs to the Desktop Info ini file. When a user with admin privileges logs in, they unknowingly run my commands. The impact of this is that if a user of Desktop Info gets "phished" by a black hat hacker, they can backdoor Desktop Info's ini file to run commands such as install malware or ransomware.

I suggest adding something to the documentation stating that the program should be placed in a secure location, such as "C:\Program Files" or "C:\Program Files (x86)". Best case would be for you to create an installer that places the program in "Program Files" by default."

There's a new secret command line option to create a pre-configured Desktop Info executable that runs without an external configuration file and so can't be easily tampered with.

Desktop Info also supports environment expansion in the log file path so if you need logging and Program Files security is preventing it, add *%appdata%* or %userprofile% or any other environment variable to the log file name in the ini file to cause the log file to be created in the user data directory or somewhere other than the Program Files path.

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 using the Run As… function.

Multi-Byte Languages

Desktop Info is able to display multi-byte languages and, to some degree, right-to-left languages. Because everything that Desktop Info displays is either stored in the configuration file or comes from Windows itself, you have full control over the language. If your language is a multi-byte language, it's important the configuration text file is UTF-8 encoded so that the text is correctly stored and read back and the font options are correctly specified.

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.

To successfully display multi-byte languages you need some or all of the following things:

- The configuration file must be UTF-8 encoded

- Specify the "font-face" option with a font capable of displaying your language. Many standard fonts in modern Windows are fine.

- Specify the "font-charset" option with a character set within the font. This will be required for some non-standard fonts.

- Specify the "font-rtl" option if your language is right-to-left

Desktop Info comes with a number of sample configuration files found in the "sample-config" directory, including some translated configurations. If you decide to use one of the sample config files, you should copy it to main directory first to avoid it being overwritten by the next new version. See the appendix for further language information.

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. Donors are acknowledged on the Special Supporters page on the web site. You can use Ko-fi, PayPal, your debit or credit card.

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

General Usage

The zip file includes several executables: in the root folder there is DesktopInfo64.exe and DesktopInfo64W.exe. These are both 64-bit builds, the first is the familiar version that attaches to the Windows desktop, the second "W" version is a regular Windows application that appears in a window with a frame, title bar, system buttons etc. They are otherwise identical. In the DesktopInfo32 folder, you will find 32-bit versions of these two executables.

This manual generally thinks in terms of the desktop attached version but everything should also apply to the "W" version.

Choose your flavour of Desktop Info and run the executable. 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.

You cannot drag Desktop Info around the desktop like a normal application, it is meant to stay anchored to the configured position. However, if you hold down the SHIFT key the drag will work. This allows you to move it temporarily and it will stay anchored in the new location. Next time you start the application, it will be back to it's configured location.

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, an image or a bar chart. The context menu can be disabled by setting the *contextmenu* item in the [options] section to '0'.

Pages	Your fully customisable pages are accessible from the context menu.
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.
Manual	Opens the manual with the default PDF reader.
Log File	Opens the log file in the default txt file application.
Export	Export the display data using the selected template.
Start W Mode	Start the DesktopInfoW.exe window mode.
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 the PAGE_TITLE key word in the [items] section of the main configuration

file. Individual menu options can be removed from the menu by removing or commenting out the option in the [text] section.

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 removing or commenting out the *menu-export* in the [text] section.

The template file can be any file type, for example HTML, XML, TXT or RTF file. There are four 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.
<dti-filter>	Optional filter.

To display just selected items, use the <dti-filter> option. This is a comma delimited list of item IDs you wish to display. For example, to display just the DATETIME item and the NETWORK WMI item, add the following to the end of the template.

```
<dti-filter>datetime,network</dti-filter>
```

See the example templates in the *sample-templates* directory. Formats like docx don't make good templates because they are compressed and aren't readily editable outside of the word processor.

Command Line Options

The following command line options are available:

/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.
/o=	Specify the section in the ini file to use for options. Overrides everything.
/debug=	Enable debug mode. Specify the full path and file name to the debug log file.
/preconfigure	The secret command line option. See Preconfigure.

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. You may optionally create a new section in the ini file called [screen-saver] and add any options to adjust the display for full screen. Some items that work in normal mode when run as Administrator may not work in screen saver mode due to reduced privileges.

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 (which is 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.
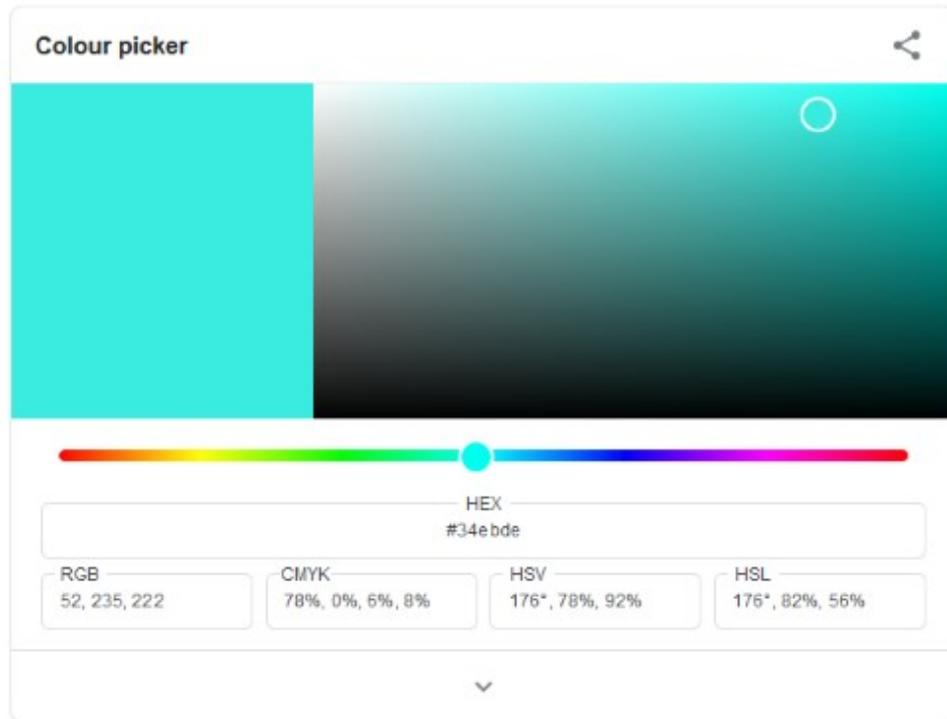
Configuration File

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.

The major sections of the configuration file are:

Options	Global settings, overall size and position, interface, logging etc
Screen Saver	Alternative options for screen saver mode.
Text	Language strings used in the context menu and elsewhere.
Functions	Predefined mathematical functions used in item expressions.
Items	The items that will be displayed.

In addition, you can define any number of custom options sections to be used instead of the normal options section.

Include Files

You can include text from other files into the Desktop Info ini file at any point by using the INCLUDE key word. This is a blind include, it does not care about context. It's up to you to ensure the included text makes sense at that point in the ini file. The external file can be any name, in any location.

```
include=mytextfile.txt
```

This will read the contents of the *mytextfile.txt* file and insert it into this position in the ini file.

```
1
2
3    [items]
4    include=colors.ini
5    COLOR=%gray%
6    DATETIME=interval:10,display:%9 %10
7
8
```

This will read the contents of the colors.ini file into at the given location.

Options Section

There are three options sections available for the three different modes:

[options]	used by the normal desktop attached mode and also acts as the default for the other modes
[options-w]	options for the DesktopInfoW.exe build. If not found, the regular [options] is used.
[screen-saver]	used in screen saver mode. If not found, the regular [options] is used.

The following tables shows all options available in the options section.

Size And Position

monitor-mode	0 = single monitor mode
	1 = virtual screen mode
monitor-num	monitor number (when 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 (default=400)
height	Total height in pixels (default=auto)
	Height is normally dynamic, this will make it fixed
display-offsetx	Offset the display from the left edge of the form
display-offsety	Offset the display from the top edge of the form
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 (default)
	0 = will effectively eliminate the column allowing the data column to occupy the whole space.
topmost	make Desktop Info window topmost
	0=Desktop Info window is attached to the desktop (default)
	1=Desktop Info window is always topmost
auto-scale	Windows 10/11 font and window scaling

0=disable (not available prior to Win10)
    (default in Win10 and later)
1=enable
toolbar   for DesktopInfoW.exe.
0=disable, 1=enable (default)

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.

Monitor Topology
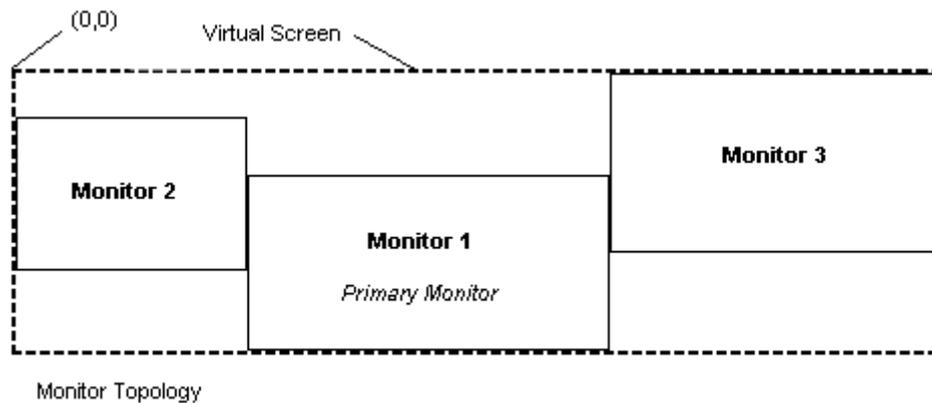
It is useful to note that if, like me, your laptop is in and out of the dock, DTI will automatically reset it's position according to the updated Windows display configuration. So, for example, I have a secondary monitor attached to the dock and configured to be above the laptop display. With DTI set 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

Auto-Scale

This can be used where your monitor pixel density is greater than 96 dpi. That is to say when you have increased the Windows display magnification to greater than 100%. By default, changing the monitor pixel density will have no effect on the DTI display, however if you set *auto-scale=1*, increasing the pixel density (magnification) will cause the DTI sizes to scale up accordingly.

It has no effect on monitors of differing pixel resolutions. For example if you set it up on a 3000x2000 monitor and then run it on a 1920x1080 monitor you can expect the apparent size to be different. In this case you should adjust the font sizes to get the desired display on the second monitor.

Interface

formcolor	background color (bgr)
transparency	background transparency. 0=opaque, 100=totally transparent (default)
contextmenu	enable the right click context menu 0=disabled, 1=enabled (default)
allowdrag	enables the ability to drag the form. 0=disabled, 1=enabled (default) Only applies to the desktop build, not the "W" build

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 *[items]* 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 *[items]* 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 font quality / anti-aliasing level
	see below for settings
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 Font Quality specifies how closely the attributes of the logical font match those of
the physical font when rendering text.

      0 = DEFAULT
      1 = DRAFT
      2 = PROOF
      3 = NON ANTIALIASED
      4 = ANTIALIASED
      5 = CLEAR TYPE
      6 = CLEAR TYPE NATURAL

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

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

PROOF: 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.

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

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

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

CLEAR TYPE NATURAL: Better than CLEARTYPE.

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.

Application Logging

Desktop Info can write to it's own text based log file as well as the Windows Application Event Log.

log	Write application logging information to the specified log file.
log-level	info=general information messages
	error=application or data errors
	warn=alarms
	debug=for tracking down application issues
	data=displayed data output

	debugonerror=write call stack when an error is raised	
	user=enable LOG items to write to the log file	
eventlog-level	Write to the Windows Application event log	
	info=general information messages	
	error=application or data errors	
	warn=alarms	
	user=enable LOG items to write to the log file	

Desktop Info performs environment expansion on the log file path so you can add environment variables such as *%appdata%* or *%userprofile%*.

The log level set in the sample config, *desktopinfo-advanced.ini* is *debugonerror*. This basically means don't log anything until there's an error. Then the previous 10 log entries leading up to the error will be logged. A bit like a call stack. This will help to trace errors with the application or configuration.

Remote Monitor Server

To start the remote monitor server, set any or all of the following options in the *[options]* section of the ini file.

server	switch on the server with default settings
	0=off, 1=on (default=0)
server-ip	switch on the server and bind to the network interface
	with this ip address, it will only respond to requests
	on this network.
	default=bind all active interfaces
server-port	the port to bind on above interfaces (default=80)
server-ssl-port	the port to bind for secure connections (default=443)
server-ssl-certificate	file name of an ssl certificate to use for secure connections

To use all the default settings, just set *server=1* to switch on the server. It will bind to all available network interfaces on port 80 for http connections and port 443 for https connections (in my case, the LAN and localhost). To use a specific network interface, just set *server-ip=x.x.x.x* to switch on the server. It will bind to just that one interface. You can also set *server-port=x* for http connections on it's own or in conjunction with the *server-ip* setting and also server-ssl-port for https connections.

In order for the server to make a https secure connection, you must have the OpenSSL DLL files available in the same directory as DesktopInfo.exe or on the search path. The two files are libeay32.dll and ssleay32.dll. These provide the encryption required to provide a secure https connection. The two DLLs in the main folder of the distribution zip file are the 64-bit versions to go

with the 64-bit version of Desktop Info. In the DesktopInfo32 folder of the distribution zip file is the 32-bit version of Desktop Info and the 32-bit versions of the two OpenSSL DLLs.

The last detail is the ssl certificate. You may supply your own certificate by adding the *server-ssl-certificate* option which contains the file name of a certificate pem file. If you do not supply your own certificate, Desktop Info will use a default internal one. While this default certificate is useful for testing, you should not consider it secure. If you test with a browser it may give you a warning about a dangerous web site!!!

The remote server logs incoming connections using the DATA logging option.

For a detailed discussion on Desktop Info remote monitoring, see <u>Remote Monitoring</u>.

Other

inimonitortime	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 log 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
share-data	This enables the local data sharing option for other instances of Desktop Info or any other application.

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 base *font-size* in the [screen-saver] options.

If you set *monitor-mode=1* then the screen saver will spread itself over the whole virtual desktop. When *monitor-mode=0*, the screen saver will limit itself to the primary monitor.

W Mode Section

This is the optional options section for the W mode, that is, DesktopInfoW.exe. It is the same as the normal options section allowing you to set different sizing and fonts etc for the W mode compared to the normal desktop mode. Because W mode is supposed to be a regular old Windows form, many of the positioning options used to anchor it in desktop mode don't apply here. However, you can set the initial width, height, center vertical and center horizontal options. These are used once to set the initial size and position of the form. After that it works like a regular old Windows form.

It has the additional option of *toolbar* to enable or disable the top toolbar.

```
[options-w]
width=450
height=600
centerv=1
centerh=1
toolbar=1
```

Text Section

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

For the menu items, if the entry is removed or commented out, 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-export	in the context menu for the "Export" parent item
	if there are no templates this item is not displayed
menu-reload	in the context menu for the "Reload" item
menu-configuration	in the context menu for the "Configuration" item
menu-log	in the context menu for the "Log" item
	if there is no log configured this item is not displayed
menu-manual	in the context menu for the "Manual" item
menu-wmode	in the context menu for the "Start W Mode" item
menu-about	in the context menu for the "About" item
menu-quit	in the context menu for the "Quit" item
nav-previous	caption for the "W" mode toolbar previous page button
nav-home	caption for the "W" mode toolbar home page button
nav-next	caption for the "W" mode toolbar next page button

Functions Section

This section allows you to declare any number of arithmetical 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 function call 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 sent from the function call.

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 display template function call. The function call from within the display template might look like this:

```
[items]
WMI= … display:{{KtoC:%temperature%}}
```

See the discussion in the [Expressions and 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:

```
3   [items]
4   COMMENT=color:%white%,style:w,font-face:Ink Free,\
5       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.

Options and Usage Topics

Key Words

Key words are words that have a particular significance and may be used anywhere in the *[items]* section. The difference between these key words and regular items is the key words are evaluated once during configuration load and then forgotten while the items are continuously evaluated. Any and all items placed inside a BEGIN-ONLOAD block are evaluated immediately during configuration load.

PAGE	Set the current page number. All following items are assigned to this page until the next PAGE.
PAGE-TITLE	Set the context menu title of the current page.
WIDTH	Sets the overall width of Desktop Info. This overrides the *width* setting in *[options]*.
BEGIN-ONLOAD	
END-ONLOAD	Items declared between these two key words are evaluated immediately during configuration load and are not saved or evaluated again. See example below.

The following key words set form level values. These override the same values set in the *[options]* section.

FORMCOLOR	Background color when TRANSPARENCY is not 100%
TRANSPARENCY	Form transparency level from 0 to 100 (default=100)
WIDTH	Form width in pixels
HEIGHT	Form height in pixels (default=auto) Height is normally dynamic, this will make it fixed
DISPLAY-OFFSETX	Offset the display from the left edge of form (default=0)
DISPLAY-OFFSETY	Offset the display from the top edge of form (default=0)

The following key words set their respective item default values. 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. You may include user variables.

FONT-FACE	Item font name
FONT-SIZE	Item font size
COLOR	Item color for right hand column or both columns if TEXT-COLOR is not set
TEXT-COLOR	Item color left column
ALIGN	Text alignment for right column 0=left, 1=right, 2=center
TEXT-ALIGN	Text alignment for left column

		0=left, 1=right, 2=center
	WIDE	Item displays over both columns.
		COMMENT is the only item that is natively WIDE.

Example 1:
```
[items]
SET Silver = #808080

PAGE=2
PAGE-TITLE=Dates and Times
COLOR=%Silver%
COMMENT=style:iwb,text:Dates / Times
…
```

Example 2:
```
[items]
SET MYFONT1 = Tahoma
SET MYFONTSIZE1 = 12

FONT-FACE=%MYFONT1%
FONT-SIZE=%MYFONTSIZE1%
COMMENT=Interesting Stuff
...
```
Example 3:
```
[items]
BEGIN-ONLOAD
# set the desktop to red to indicate some state
# get the current monitor specs
MONITOR=set:sx=%4 sy=%5
# make a decision and set the new values
IF=value1:2,value2:1,comparator:eq,eval:6
width={{%sx%-100}}
height={{%sy%-100}}
display-offsetx=1300
display-offsety=0
formcolor=0000f0
transparency=60
END-ONLOAD
...
```

FONT-SIZE can be an absolute value or relative value. Relative values are given as a percentage of the last absolute value, the base value. This allows you to set a base font size once and then use relative values to scale it up and down as required for different items. When a percent is specified, it is always the percentage of the last absolute value given, so that FONT-SIZE=100% will return the font size to the base value.

```
 1
 2
 3   [items]
 4   # set base font size
 5   font-size=10
 6   # grow font size by 150%
 7   font-size=150%
 8   # return to the base size of 10
 9   font-size=100%
10
11   |
```

The BEGIN-ONLOAD block is useful for gathering some initial information and setting variables, fonts or other parameters before anything appears on the screen. For example, check the screen resolution and set the font size accordingly.

```
 1
 2
 3   [items]
 4   BEGIN-ONLOAD
 5       MONITOR=set:hres=%4,hidden:1
 6       # start with default size 10
 7       FONT-SIZE=10
 8       # if hres is greater than or equal to 1600
 9       IF=value1:%hres%,value2:1600,comparator:ge
10       FONT-SIZE=12
11   END-ONLOAD
12   |
```

```
 1
 2
 3   [items]
 4   BEGIN-ONLOAD
 5       # adjust the font size and width for high res display
 6       MONITOR=set:hres=%4,hidden:1
 7       # start with default size 10
 8       FONT-SIZE=10
 9       WIDTH=440
10       # if hres is greater than or equal to 1600
11       IF=value1:%hres%,value2:1600,comparator:ge
12       FONT-SIZE=12
13       IF=value1:%hres%,value2:1600,comparator:ge
14       WIDTH={{Round(%hres% / 4)}}
15   END-ONLOAD
16
```

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.

key	value
hidden	0=not hidden (default)

	1=item is 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.
color	A bgr or rgb value as described in the Colors section. This overrides the current default COLOR value. If *text-color* is not specified then this controls the color for both columns. May use variables and expressions.
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 *color* option. May use variables and expressions.
style	Font style
	b=bold, i=italic, u=underline, w=full width underline
csv	output item 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 section. 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
alarms	any number of alarms
chart	Display a chart as described in the Charts 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
wide	stretch column 2 display over both columns
text	template for the left column display. See discussion here.
align	Align both columns text. Overrides the global setting.
	0=left align (default), 1=right align, 2=center align
text-align	Align the left column title text. Overrides the global setting and any ALIGN setting.
	0=left align (default), 1=right align, 2=center align

text-offset	Offset the title text in pixels from the left margin (or the right if you have right align set).	
row-text	Provide a title text in the left column for each logical row in a multi-row item. Alternative to *text* 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 in a number of places such as remote monitoring. By default, every item is automatically assigned a unique id based on the item type and possibly followed by a number, such as *comment, comment1, comment2* etc.	
onload	This allows an individual item to be evaluated during configuration load and also saved to the active config in the normal way as opposed to items in the BEGIN-ONLOAD block which are **not** saved and hence never evaluated again.	
global	this item will appear on every page at the given position useful for banner images, messaging, navigation >=0 appears on every page at this position where 0 is the topmost position on the page -1 appears on every page at the bottom	
multirow	DTI uses the pipe symbol '|' to denote a new line. You can disable this for any given item by setting multirow:0.	
no-wait	runs the item in it's own thread. Use this to return immediately and prevent the display being blocked by long running items. Use the *executing* option in the *[text]* section to adjust the message. See thread discussion below for more information. 0 = wait (default) 1 = no wait with executing message 2 = no wait without executing message	
regex	a regular expression applied to all return values before they are inserted into the display row or assigned to a variable. See the discussion below.	
regexr	the expression used to replace characters matched by the above regex expression. If not specified then matching characters are simply removed.	

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

Background-interval: Set this when you want an item to continue collecting even when it is not on the current page. This is useful to keep charts ticking over and csv data files writing or if you're monitoring remotely. 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 the item data is collected at the *background-interval* rate.

Hidden: Set the *hidden* option if you want the item to be collecting data but not visible on any page.

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 enabled, hide the ethernet item if ethernet is not connected. The item continues to collect 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 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.

Threaded Execution: Some items can take a long time to evaluate, like the Windows Updates query or the TPM query. While these are executing, the whole application is blocked until they complete. The solution is to run these queries in their own thread so that Desktop Info can continue to run other queries and keep the display ticking over without being blocked by one long query. When the long query has completed the results will be displayed in the usual manner.

To do this, add the *no-wait* option to the item. Set the value to '1' to show an *executing* message, or '2' to hide the item until the results are available.

```
CMD= …… , no-wait:1
```

The item will return immediately without waiting for results. During the next cycle through the items, DTI will check to see if the thread is finished. If it is, the results are retrieved and the thread is destroyed.

The '*executing*' string can be added to the [text] section. There is a cost to creating and destroying many threads so best reserve this option for only long running items.

Regular Expressions: Use the *regex* and *regexr* common options to massage the raw data before it is inserted into a display row or user variable. This allows you to do things such as trim leading/trailing white space or remove pipe characters. *Regex* is the expression used to match

characters and *regexr* is the expression used to replace the matched characters. If *regexr* is not specified then the matched characters are simply removed. The regex parser is the open source PCRE library as shipped with Delphi. There is an example in the PHYSICALDRIVES item in the desktopinfo-advanced.ini file.

I'm not the regex guru so don't ask me to write your regex expressions. There is a good tutorial at: www.regular-expressions.info.

Examples:

Trim leading and trailing white space:

```
^[ \t]+|[ \t]+$
```

Remove pipe characters:

```
\|
```

Trim leading and trailing white space and remove pipe characters:

```
^[ \t]+|[ \t]+$|\|
```

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". There is also an alternative syntax which might be more familiar to some:

```
set MyTestKey1=MyTestValue1
```

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.

```
3  [items]
4  HOST=set:hostname,hidden:1
5  USER=set:username,hidden:1
6  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.

```
3  [items]
4  USER=set:username,hidden:1
5  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.

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

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

```
1
2
3   [items]
4   SET=key:var1,value:complete
5   SET=key:var2,value:output
6   SET=key:var3,value:%var1% %var2%
7   TEXT=display:%var3%
8
9
```

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.

```
1
2
3   [items]
4   SET white = #ffffff
5   SET cyan  = #00ffff
6   SET green = #00ff00
7   COMMENT=color:%white%, text:Desktop Info
8
9
```

Note the alternative form of the SET item.

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.

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%
```

Alarms

Alarms allow a variety of ways to examine the data and also allow you to compare the current data against the previous dataset so it can detect changes in data. They will also compare any two values and/or expressions whether or not they come from the current dataset.

Any item that displays a numerical value can be configured to change color when the data meets a given criteria.

The general format is:

```
alarms:(%cur% cmp %pre% [%col%]) (%cur% cmp %pre% [%col%]) ...
```

where

%cur%	may be a return value from the current dataset or any other value or expression
%pre%	may be a return value from the previous dataset or any other value or expression
cmp	a comparator (same as IF item)
%col%	the color the item will become when the comparison is true

The color is optional. If the *%col%* option is set, the display will change color. If the *%cur%* option is a return value, the display line containing that return value will change color. If it's not a return value (eg a number or expression) then the first display line of the item will change color. If the *%col%* option is not set, there is no visual indication of the alarm. If the WARN option is set in the *log* option in *[options]*, an entry will be written to the log.

The parser separates the options based on space so if you use an expression you should not include spaces.

You may define as many alarms as you want for each item. The last one to set the color wins.

Some Examples:

If 1 equals 2 then go red (literally, not return values)

```
alarms:(1 eq 2 %red%)
```

If latest return value %1 is greater than 50 then go blue

```
alarms:(%1 gt 50 #0000f0)
```

If latest return value %2 is different to the previous return value %2 then go orange

```
alarms:(%2 ne %2 %orange%)
```

If %caption% contains Microsoft then go green

```
alarms:(%caption% contains Microsoft #00f000)
```

If 100 is greater than 10x10 then go green

```
alarms:(100 gt {{10*10}} %green%)
```

In this example, the time will turn red for 10 seconds whenever the minute ticks over:

```
DATETIME=interval:10,display:%9 %10,alarms:(%10 ne %10 %red%)
```

Charts

Any numerical value can be charted. You can select from several different styles with a linear or logarithmic scale. All the charts are interchangeable and can easily be switched from one to the other and back again.

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

```
chart:style option:value option:value option:value ...
```

The *style* value can be: bar, line, vbar, vbarfilled, scatter, filled, 3d.

Where *bar* is the legacy horizontal bar chart. Following the *style* 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 horizontal bar charts.

Options:

style	bar, line, vbar, vbarfilled, scatter, filled, 3d
scale	linear or log (default is linear)
min	minimum value of the chart
	is always zero for horizontal bar chart
	optional for others, if not specified, the vertical scale will
	automatically adjust based on the current minimum value
max	maximum value of the chart
	required for horizontal bar chart
	optional for others, if not specified, the vertical scale will
	automatically adjust based on the current maximum value
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 (default=green)
color2	the color of the second line series
	or the medium value bar color (default=orange)
color3	the color of the third line series
	or the high value bar color (default=red)
threshold	the threshold value to transition from medium color to
	high color of the bar chart (default=none)
width	the width of the chart in pixels (default=auto size)
height	the height of the chart in display rows (default=1)
points	the number of data points on the chart (except horizontal bar)
	this has the effect of stretching or compressing it
	or creating a static chart with points:1
	the default is one point per pixel
border	border visibility (default=1)
row-id	the return value used to identify a row for non-bar
	charts (required for multi-row line charts)

The value to be charted can be specified a number of ways such as return value number, name, user variable or expression. 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 horizontal bar chart can display one or two series. All the others 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 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.

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:line max:100 series1:1
```

Bar Chart Example:

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

```
chart:bar max:100 series1:1
```

Expression Example:

A simple line chart which charts an expression:

```
chart:line max:1000 series1:{{%1*10}}
```

Series Examples:

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

```
chart:line 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 Desktop Info, 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:bar max:100 series1:%EstimatedChargeRemaining% threshold:-40
```

The Zero Line And Mirror Series

By default, if a series contains negative values, the zero line will move up from the bottom of the chart so that the negative values can be displayed. As the maximum and minimum values of the chart change, so the zero line will adjust to fit. If you want to keep the zero line fixed to a certain position then set the *min* chart option.

If you want to do a mirror series, where the first series is above the zero line as per normal and the second goes below the zero line, just add the *Neg()* function to the second series value:

```
chart:line series1:%1 series2:{{Neg(%2)}}
```
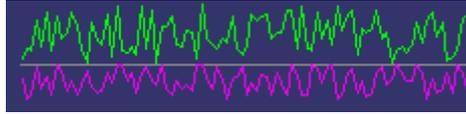
Chart Positioning

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

```
… chart:bar max:100 series1:%7,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 max:100 series1:1,display:%1 %chart%
```

This works well for CPUUSAGE for example by placing the bar chart immediately after the value on the same display line.

You may place additional 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 max:100 series1:1,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.

Remote Monitoring

To monitor a remote instance of Desktop Info, the first thing to do is switch on the remote monitor server of Desktop Info on the machine you wish to monitor. Just set the *server=1* setting in *[options]* to use the defaults.

This will bind the server to all available network interfaces on port 80 and port 443. You can limit it to binding to a single interface by setting the *server-ip=x.x.x.x* setting and set a different port using the *server-port=x* and *ssl-server-port* settings. If the server fails to start then either the ip address is not valid or one or both of the ports are in use. You might try different ports, say 8080 or 8000 for http and 4443 for https. This is all you need to do to get the remote monitor server running.

Note: For secure connections you need to have the two DLL files, *libeay32.dll* and *ssleay32.dll*, present in the directory with the Desktop Info executable. The DLL files in the root directory of the distribution zip file are 64 bit while the DLL files in the DesktopInfo32 directory of the distribution zip are 32 bit.

Test the server is listening by opening a browser on the same machine and typing:

> http://localhost

You should get a response back from Desktop Info. Now, assuming Desktop Info is displaying the CPU item, type in the following data request:

> http://localhost/data?id=cpu

You should get a response containing an xml snippet:

```
-<desktop_info version="3.1.0">
  -<raw_values>
    -<row>
        <data name="cpu" type="3">2</data>
        <data name="kernel" type="3">1</data>
        <data name="queue" type="6">0</data>
    </row>
  </raw_values>
</desktop_info>
```

By default, every item is assigned a unique id during startup. This unique id is normally the same as the item name but if you have multiple items then subsequent ones will have a number appended. You can override the default unique id by setting the id option on the item. If you want to monitor items regardless of the visible page, you will also need to set the *background-interval* option.

Now that the server is working you just need to add a DTI item to your client instance of Desktop Info (the instance that is retrieving the data for display) for each item you wish to retrieve. In the item options, specify the unique *id* that identifies the item you want to retrieve.

```
1
2
3   [items]
4   DTI=host:10.0.0.34,id:cpu,interval:10,display:Cpu: %1%
5
6   |
```

This item connects to Desktop Info on the host 10.0.0.34 and retrieves data for the *cpu* item. You are now free to use the display template, text and row-text to format and display the raw data as desired, add alarms and maybe a chart.

WMI items can use their property names such as as %CurrentTemperature% etc.

External Collectors

Desktop Info can display data collected by an external collector. A collector is a stand alone application. You can use any tool at your disposal to write a collector that's capable of collecting the data you need, constructing XML and writing it to a Windows shared memory area. The collector implementation details are left to the author. An example Delphi collector is available on the web site including details on the XML format and supported data types.

Once you have a collector, you simply add a COLLECTOR item to your config file. The *name* option is the name of the shared memory area to which your collector is writing. That's it. Desktop Info will attempt to open the shared memory and if successful, read the XML and convert it to data. Any errors in reading the data are noted in the log when *error* or *debugonerror* is enabled.

```
[items]
COLLECTOR=name:SampleCollector_Item1, interval:1, display:%1 %2 %col3%
COLLECTOR=name:SampleCollector_Item2, interval:1, display:%1
```

Local Data Sharing / Multiple DTI Displays

Desktop Info can share it's data locally by switching on the *share-data* option in the *[options]* section of the ini file. When this option is enabled, Desktop Info writes each item's data in the form of XML to a Windows shared memory area using the id of the item as the name of the shared area. Other applications are now free to retrieve this data for their own purposes. Also you can run additional instances of Desktop Info with their own ini file to retrieve this data and display it. This provides a mechanism for displaying a subset of DTI data on a secondary monitor for example. Because the secondary instance is not collecting data itself, the cpu overhead is very low.

In the ini files of the secondary instances of Desktop Info, use the *COLLECTOR* item to retrieve data from the shared areas. This treats the main DTI instance as if it were an external collector.

Of course, the secondary instance is free to collect it's own data but if the main instance is already collecting that data then that would be the preferred option to reduce cpu overhead.

```
40    └
41  ⊟ [items]
42    ╎ collector=name:user,interval:10,set:username,hidden:1
43    ╎ collector=name:host,interval:10,text:Host/User,display:%1/%username%
44    ╎ collector=name:local_time,interval:5,text:Date/Time,display:%11
45    ╎ collector=name:coretemp,interval:20,interval:1,text:Core Temp,display:%1%2 (%5)
46    ╎ collector=name:cpu,interval:2,text:Cpu,chart:filled series1:1 color1:00ff00 series2:2 color2:b
47    ╎ collector=name:latency,interval:10,id:latency,font-size:120%,alarms:(%11 ge 100 0088f0),text:N
48    ╎ collector=name:alarms,interval:5,text:Last Alarm: %10, display:%14,maxrows:1,hide-no-result:1
49    ╎ collector=name:osbuild2,text:OS Build,display:%1{{{ (%2)}}}
50    └
```

Preconfigure

You may wish to deploy Desktop Info to computers that are used by other people, for example in a corporate environment or a testing environment but you don't want other people to see and/or edit the configuration file. You would prefer to set it once and know that it can't be seen or altered by other people. This is when you use the *preconfigure* option.

Preconfigure is the act of creating a new copy of the Desktop Info executable with the configuration file embedded. In this way you can deploy just the executable, with no other files, and it will be pre-configured the way you want.

NOTE: This tool is not perfect. The output executable will lose the digital certificate and if you try to sign it yourself using signtool, the embedded ini file will be trashed.

To preconfigure Desktop Info, you need to go to the command line in the Desktop Info directory. The command line is:

```
desktopinfo.exe /preconfigure
/ini=preconfigured.ini /out=mynewdesktopinfo.exe [/silent]
```

The */ini* argument is whatever configuration file you want to embed. The */out* argument is the name of the executable file that will be created. The */silent* argument suppresses the message dialog boxes.

The new executable is a copy of the current executable with the pre-configured ini file embedded. When Desktop Info starts, it will look for this embedded ini file. If it finds it, it will be used and all other options are ignored. If it's not found then it will proceed in the normal manner.

If it detects the embedded configuration has been tampered with it will exit immediately. You can not embed multiple configurations into an executable.

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 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 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 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 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"

Formatting Strings

Well, we can pad them to help make table-like displays. This is most useful when combined with a fixed width font. The general form is:

```
display:%1[w.ps]
```

Where 'w' is a number that represents the maximum allowed size of the string in characters. If it is longer, it is truncated, if it is shorter, it is **left** padded with spaces.

And 'p' is a number that represents the maximum allowed size of the string in characters. If it is longer, it is truncated, if it is shorter, it is **right** padded with spaces.

Example:

```
display:%1[10.0s] %2[6.0s] %3[6.0s]
```

See the *sample-config\desktopinfo-linux.ini* configuration file for an example.

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 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 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 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".

If that was a decimal transform:

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

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

or

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

Displays the 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 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, boolean values 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 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 and Functions

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 from Kelvin to Celsius or Fahrenheit or to display a percent value calculated from two return values. If there is any part of the expression that cannot be evaluated (ie illegal characters) or the expression is somehow incomplete, 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 definition immediately following the closing curly braces.

```
{{expression}}[1.0f]
```

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

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

will result in the display, '3.0'.

Remember, in order to qualify as a display format definition, the opening square bracket must be the very next character after the second closing curly brace.

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.

```
1
2
3  [items]
4  WMI=interval:10, text:Cpu Temp, namespace:root\wmi, \
5      query:MSAcpi_ThermalZoneTemperature, \
6      display:{{%CurrentTemperature%/10-273.15}}[1.0d]Celsius
7
8
```

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

```
 9
10
11  [items]
12  WMI=interval:10, text:Cpu Temp, namespace:root\wmi, \
13      query:MSAcpi_ThermalZoneTemperature, \
14      display:{{ (%CurrentTemperature%/10-273.15) *9/5+32}}[1.0d]Fahrenheit
15
16
```

So, in summary, a normal expression looks like this:

```
{{10 * 5 + 8 / 2}}
```

A normal expression with optional format definition looks like this:

```
{{10 * 5 + 8 / 2}}[2.2f]
```

Mathematical Functions

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, Neg, Power, RadToDeg,
RadToGrad, Rnd, 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)}}
```

See the chart demonstration in the *sample-config\desktopinfo-advanced.ini* file for some real world examples of expressions containing mathematical functions.

Functions

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.

The general format for calling a function is:

```
{{function:arg1[:arg2:…]}}
```

Where *function* is the name of a function defined in the [functions] section of the ini file, 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 the item's 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.

```
 2
 3
 4   [functions]
 5   KtoF=(%1/10-273.15)*9/5+32
 6
 7
 8   [items]
 9   WMI=interval:10, text:Cpu Temp, namespace:root\wmi, \
10       query:MSAcpi_ThermalZoneTemperature, \
11       display:{{KtoF:%CurrentTemperature%}}[1.0d]Fahrenheit
12
13
```

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. The display format definition follows the function call.

So in summary, a normal function call looks like this:

```
{{name:10:5:8:2}}
```

A normal function call with optional format definition looks like this:

```
{{name:10:5:8:2}}[1.2f]
```

Hide Specific Output

The display template of each item controls how and what data is displayed. In some situations the displayed data becomes <null> because there is no data, either the WMI query failed or some return value is not available. The actual string displayed for null data is configured using the *null-result* string in the *[text]* section of the ini file.

It is possible to instruct Desktop Info to not display some data if it is not available. This is called the Not Null Expression. Using triple curly braces, you define a section of the display template that

should only display if all the return values within have a non-null value. Or, to put it another way, if any of the return values in this section are null then do not show the whole section.

For example:

```
display: %1 %2 {{{Show %3 and %4 only if both are not null}}}
```

Return values %1 and %2 will always display but the string "*Show %3 and %4 only if both are not null*" will only display if both %3 and %4 have legitimate values. If either one is null then the whole string is skipped.

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 alarms. Use the item return value numbers for the chart and alarm series.

ALARMS

Displays the most recent alarms generated by the *alarms* option of each item.

Available Options:

maxrows	maximum number of alarms to display

Return Values:

%1	%dow%	day of the week
%2	%day%	day
%3	%month%	month
%4	%year%	year
%5	%hour%	hour
%6	%minute%	minute
%7	%second%	second
%8	%unix%	unix time
%9	%short_date%	short date
%10	%short_time%	short time
%11	%long_date%	long date
%12	%long_time%	long time
%13	%alarm_text%	text (left column)
%14	%alarm_display%	display (right column)

Default Display:

```
display:%1 %2
```

See Also:

Common Options

Alarms

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

Default Display:
```
display:%1
```

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.

See Also:

[Common Options](#)

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

Default Display:
```
display:%9 %10
```

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.

See Also:

[Common Options](#)

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 *errorlevel* 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.

Default Display:

```
display:%4
```

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 maxmimized
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
```

Parameters

The *parameters* option accepts user variables. You can use the <u>SET-SECTION</u> item to load a [section] of the ini file into a user variable which can then be used in the CMD *parameters*. The *key* option is the name of the user variable and the *value* option is the name of the ini [section] that will be loaded into the variable.

```
 5
 6  [items]
 7  SET-SECTION=key:rdp-users,value:rdp-users.ps1
 8  CMD=interval:60,text:RDP Users,trim:1,file:powershell.exe,parameters:%rdp-users%,code-page:437
 9
10
11  ...
12
13
14  [rdp-users.ps1]
15  $output = $(quser) -Replace '>',''
16  $output = $output -Replace '\s{2,}','`t'
17  $($output.trim() | convertfrom-csv -delimiter '`' | ft -Property USERNAME -HideTableHeaders| Out-String).Trim()
18
```

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%"
```

See Also:

Common Options

COLLECTOR

Retrieves data from an external data collector (or another instance of Desktop Info). See the External Collectors section for more information on implementing external collectors and also the discussion and example collector on the web site.

Available Options:

name	name of the shared memory area to retrieve external data
	the id of the main DTI item
maxrows	the maximum number of rows to display (default=1)

Return Values:

referenced using %1 or %column_name% style syntax

Default Display:

```
display:%1
```

Desktop Info will attempt to open the shared memory area given by *name* and if successful, read the XML and convert it to data. Any errors in reading the data are noted in the log when *error* or *debugonerror* is enabled.

The return values are whatever the collector passes over and you can display them as you would any other DTI data. You can refer to them using *%1* or *%column_name%* style syntax.

This is also the mechanism for creating multiple Desktop Info displays on the same computer but, perhaps, on different monitors. Enable the *[share-data](#)* option in the options section for the first/main instance of DTI then in the secondary instances use the COLLECTOR item to retrieve the data.

```
1
2
3    [items]
4    COLLECTOR=name:SampleCollector, interval:6, display:%1 %2 %col3%
5    COLLECTOR=name:SampleCollector, interval:6, display:%1
6
```

Be aware that if the main DTI instance stops sharing data for any reason, the secondary instances will continue to display the last available data set.

See Also:

[Common Options](#)

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/display	Text to display over both columns

Return Values:
%1	comment text

Default Display:
```
display:%1
```

This is a simple text display item most suitable for banners, section headers, messages and other random text comments. It is *wide* by default. In it's simplest form of *"COMMENT=message"*, the given text will be displayed over both columns. The second form which is functionally equivalent is *"COMMENT=text:message"* or *"COMMENT=display:message"*. In this case the result is the same, the given text is displayed over both columns. If both *text* and *display* are specified then the *display* text is displayed over both columns. If you set *wide=0* then the *text* and *display* options are displayed in the left and right columns respectively as you would expect.

See Also:

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.

Available Options:

type	0=button (default), 1=hyperlink
color	text color
button-color	the background color of the button **or**
	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)
height	fixed control height in pixels (default is auto-sized)
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: *style* and *control-style*. If neither 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 just the control 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%
```

Width and height can be set for both hyperlinks and buttons. In both cases, if not specified, they are calculated from the displayed text. You still need to set the left option. The height of the display row is set by the height of the biggest control. Both controls will word break where the control is not wide enough.

See Also:

[Common Options](#)

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

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 is ignored.

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 retrieve the data. 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 given computer than the WMI method.

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

Hint: There is a zip file, otherwise pay attention during the installation process!

Return Values:

%1	highest core temp
%2	scale (F or C)
%3	core count
%4	cpu count
%5	cpu name

Default Display:
```
display:%1
```

See Also:

[Common Options](#)

CPU

Shows overall 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
%4	number of processes

%5	number of threads
%6	system calls per sec
%7	context switches per sec
%8	exception dispatches per sec

Default Display:
```
display:%1
```

See Also:

[Common Options](#)

CPUCOUNT

Shows the number of cpu cores in the system.

This item returns two values that are theoretically the same but derived from different sources. The second requires the Processor performance counter. If this counter is not available, the second value item returns no results.

Return Values:

%1	core count derived from GetSystemInfo (kernel32)
%2	core count derived from Performance Data Helper API

Default Display:
```
display:%1
```

See Also:

[Common Options](#)

CPUTYPE

Shows information about the brand and type of cpu.

Return Values:

%1	cpu information

Default Display:
```
display:%1
```
See Also:

[Common Options](#)

CPUUSAGE

Shows the usage for each cpu 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 cores to display
filter	list of cores to display separated by a space
	(eg. "0 3 6 63")

Return Values:

%1	core number
%2	core usage

Default Display:
```
display:%1 %2
```
See Also:

[Common Options](#)

DATADUMP

Writes a csv data file for every collecting 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 the item 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 items written
%2	date/time of last data dump

Default Display:
```
display:%1
```
See Also:

[Common Options](#)

DATETIME

The current date and time of the host or the current date and time of some other time zone.

Available Options:

offset	the offset of the desired time zone in minutes
	(where 0=UTC time, omit this option for the local time)

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

Default Display:
```
display:%9 %10
```

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.

To display the time for some other time zone add the *offset* option. The offset is the difference in minutes from UTC time for a particular place. To work out the offset minutes just take the time zone offset hours and multiply it by 60 and add the minutes. For example, the UTC offset for Sydney is +10:00 so 10 * 60 is 600 minutes. The UTC offset for Adelaide is +09:30 so 9 * 60 + 30 is 570 minutes. New York is -04:00 so -4 * 60 is -240 minutes. Note the negative numbers there. Honolulu is -10:00 so -10 * 60 is -600 minutes. This is the number you put in the offset option. This doesn't account for daylight saving time so you'll have to adjust it on the appropriate dates.

See Also:

Common Options

List of Time Zones: https://en.wikipedia.org/wiki/List_of_tz_database_time_zones

DIRECTX

Shows the Direct X version.

Return Values:

%1	Direct X version

Default Display:
```
display:%1
```
See Also:

Common Options

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

Default Display:
```
display:%1
```
See Also:

Common Options

DOMAIN

Shows the current domain.

Return Values:

%1	domain name

Default Display:
```
display:%1
```
See Also:

Common Options

DOMAINCONTROLLER

Shows the current domain controller.

Return Values:

%1	domain controller

Default Display:
```
display:%1
```
See Also:

Common Options

DTI

Retrieves raw data from a remote instance of Desktop Info. The remote instance must have the remote monitor 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, alarms, row-text, charting etc.

Available Options:

host	The remote host. This can be a host name or ip address. May include protocol (default=http) and port (default=80). If you specify http, the default port is 80. If you specify https, the default port is 443.
timeout	connection/read timeout in milliseconds (default=3000)
id	the item id for the item you wish to retrieve.
maxrows	the maximum number of rows to display (default=1)

Default Display:
```
display:
```

The return values are the same as the remote item you are retrieving. WMI items may use their property names. If you get a <n/a> response, it means the connection to the server timed out. If you're using the default timeout value, it will probably just pick up it next time around. You can up the timeout value to reduce the connection timeout failures to a slow server but be aware this will block all data collection on the client until it returns.

See the discussion at Remote Monitoring for a detailed explanation of setting up remote monitoring.

```
1
2  [items]
3  DTI=host:10.0.0.15,id:cpu,interval:10,display:Cpu: %1%
4  DTI=host:10.0.0.34,timeout:500,id:temp,interval:10,\
5   display:{{1.0f:%CurrentTemperature%/10-273.15}}C / {{1.0f:%CriticalTripPoint%/10-273.15}}C
6
7
```

See Also:

Common Options

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

Default Display:
```
text:%1, display:%2
```

Example:

```
envvar=key:SystemDrive
```

See Also:

[Common Options](#)

EVENTLOG

Returns the most recent entries for the given event log.

Available Options:

log	name of the event log to retrieve (application, system, security etc)
maxrows	maximum number of entries to display
filter	filter on the source field

Return Values:

%1	date/time generated
%2	event id
%3	event type
%4	task category
%5	source name
%6	computer name
%7-%10	first four event data entries

Default Display:
```
display:%1
```

The following example will display the most recent entry from the system event log:

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

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

Example:

```
filter:+service-manager
```

Includes all entries that contain the keyword "service" and excludes entries that contain the keyword "manager" in the source field.

If there is more than one filter match, they are evaluated left to right and the last one wins. An entry must match all the given filters in order to be displayed.

See Also:

[Common Options](#)

FILE

Displays various information about files and directories. Can be local or network. If specifying a directory, make sure it does not have a trailing backslash.

User variables and environment variables may be used in the file/folder name option. The create time, last write time and last access time displays are formatted using the local system global short date format and long time format.

Available Options:

        file   file or folder name

Return Values:

%1	the file/folder name
%2	size in bytes
%3	file version
%4	create time
%5	last write time
%6	last access time

Default Display:

```
display:%1
```

See Also:

[Common Options](#)

FILE2TEXT

Displays the contents of the given text file containing 'key=value' pairs. The file may contain any number of lines of text. Each line in the file that contains an equals '=' sign generates a row of data. So, many lines in the file will result in many rows on the screen.

You can filter the lines from the file by using the 'key' option. Set the interval if you want to update it at regular intervals. The default display template is "%1 : %2" but, of course, you can put anything you want. User variables may be used in the file name option and key option and also within the text of the file.

Supports DOS (CRLF) and Unix (LF) text files and multi-byte Unicode text (UTF-8).

Available Options:

file	name of file containing *key=value* pairs
key	display only the value given by key

Return Values:

%1	the key (text before the equals sign)
%2	the value (text after the equals sign)

Default Display:

```
display:%1 : %2
```

Example:

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

     [Common Options](#)

FILECONTENTS

Displays the contents of the given text file. Each line of text generates a row of data. So, many lines in the file will result in many rows on the screen. Set the interval if you want to update it at regular intervals. The default display template is "%1". You can change the display template to include other text or produce an indent or whatever.

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 Unicode text (UTF-8).

Available Options:

file	name of file to display
top	display the first x lines in the file
tail	display the last x lines in the file
wide	0=display in right column (default)
	1=display over both columns

Return Values:

%1	a line of text

Default Display:
```
display:%1
```

The top and tail options can be used together or independently to show the first lines in the file and/or the last lines in the file. If neither is specified, the whole file is displayed.

Example:

```
FILECONTENTS=file:instructions.txt,wide:1
```

To show the first line of the file:

```
FILECONTENTS=file:instructions.txt,top:1
```

To show the last 5 lines of the file:

```
FILECONTENTS=file:instructions.txt,tail:5
```

See Also:

[Common Options](#)

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

Return Values:

%1	file data

Default Display:

```
display:%1
```

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 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
```
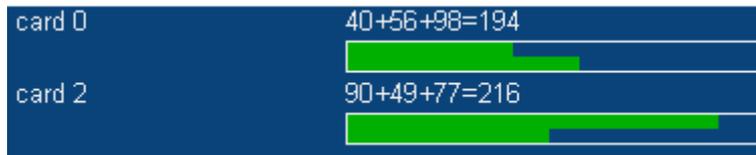
See Also:

> Common Options

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

Default Display:
```
display:%1
```

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!"
```

See Also:

> Common Options

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

Default Display:

```
display:%1
```

See Also:

[Common Options](#)

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

HTTPGET

Makes a HTTP/S 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.

Note: For secure connections you need to have the two DLL files, *libeay32.dll* and *ssleay32.dll*, present in the directory with the Desktop Info executable. The DLL files in the root directory of the distribution zip file are 64 bit while the DLL files in the DesktopInfo32 directory of the distribution zip are 32 bit.

User variables may be used in the source option.

Available Options:

     source   host url

     timeout   http connection/read timeout in milliseconds (default=3000)

Return Values:

     %1   returned text from http host

Default Display:

```
display:%1
```

```
1
2  [items]
3  HTTPGET=interval:600,source:http://plain-text-ip.com/
4
5  |
```

If you need to make a connection to a host that's specifically ipv6, you must enclose the host name part of the url in square brackets thus:

```
http://[ipv6.plain-text-ip.com]
```

The supported protocols are: TLS v1.0, TLS v1.1 and TLS v1.2.

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/

See Also:

     Common Options

IEVERSION

Shows the Internet Explorer version.

Return Values:

     %1   ie version

Default Display:

```
    display:%1
```
See Also:

    [Common Options](#)

IF

The IF item has two distinct ways of operating and a murky, gray area in between.

Basically, it will compare two values using a comparator and return a result. If both values are valid numerical values, they are compared as numbers, if not they are compared as character strings.

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

In it's simpler form, the return value will contain the outcome of the comparison (True or False) and, if the outcome is True, the following item is evaluated or, if the outcome is False, the following item/s is skipped.

In it's extended form, the return value can be anything you specify for the True and False outcomes and, depending on whether that return value can be interpreted as a boolean (ie True or False), it may or may not evaluate the following item/s. Easy, right?

As with all items, you can assign the return value to a user variable to be used later. IF and SET are the only two items that will always be evaluated regardless of the currently displayed page. In other words they are global.

Available Options:

value1	the first value to compare	
value2	the second value to compare	
comp	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
true	assign this value to the result if the outcome is true	
false	assign this value to the result if the outcome is false	
eval	the number of following items to evaluate if the result is TRUE, otherwise skip those items, default=1	

Return Values:

       %1   this will be a basic True or False result **or** one of the *true* or *false* options

Default Display:

```
display:%1
```

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, comp:eq
```

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

```
1
2
3    [items]
4    # get the OS build
5    OSBUILD=set:OSBuildVer,hidden:1
6
7    # if it's Win 10 1903/1909
8    IF=value1:%OSBuildVer%, value2:6.3.18, comp:contains
9    CONTROL=text:Support:,display:Helpline,uri:https://intranet/help-win19
10
11   # if it's Win 10 1803/1809
12   IF=value1:%OSBuildVer%, value2:6.3.17, comp:contains
13   control=text:Support:,display:Helpline,uri:https://intranet/help-win18
14
15
```

The extended version allows you to specify the true or false return values and assign the outcome to a user variable.

```
1
2
3    [items]
4    IF=value1:100, value2:1000, comp:eq, true:They Are Equal, \
5      false:They Are NOT Equal, set:res=%1
6    COMMENT=%res%
7
```

In this case, the result can't be interpreted as a boolean (ie true or false, 1 or 0) because the return value will be either "They Are Equal" or "They Are NOT Equal", so it will always evaluate the following item. At the end of the IF item, the return value %1 is assigned to a user variable and displayed in the following COMMENT item.

So the general rule is if the return result is True or False, 1 or 0 then it can decide whether to evaluate the following item/s. If the return result is not one of these then the following item/s will always be evaluated.

By default, when the IF item returns a FALSE result, the next item is skipped. You can skip any number of following items using the *eval* option. For example, set *eval:2* to skip the following two items if the result is FALSE. Conversely, think of it as, *"if the result is TRUE evaluate the following 'n' number of items"*.

In addition, you can add the *onload:1* option to perform one-off evaluation during configuration load.

See Also:

> Common Options

IMAGE

Displays an image file. The image can be jpg, tif, png, gif, bmp, wmf, emf format. By default, the image is displayed in the right column only but you can display it in the left column or over both columns by setting the *column* or *wide* options. If the image is in the right column only, you can use the *text* option to add text to the left column. If the image is in the left column only you can use the *display* option to add text to the right column.

Supports user variables in the *file* option. Starting with v3.10.1, IMAGE is evaluated every cycle according to the *interval* setting. This allows you to change the image on the fly.

Available Options:

file	the image file to display
align	horizontal alignment
	-1= align left
	0=align centre (default)
	1=align right
offset	offset from left or right in pixels when *align* is used
width	stretch the image horizontally
	0=maintain original width or adjust for aspect (default)
	>0=stretch to this width in pixels
	-1=stretch to form width
height	stretch the image vertically
	0=maintain original height or adjust for aspect (default)
	>0=stretch to this height in pixels
aspect	keep original aspect ratio
	0=do not keep aspect ratio
	1=keep aspect ratio (set only one of width or height) (default)
column	put the image in left column or right column

<div style="text-align:center">

0=left column
1=right column (default)

</div>

Default Display:
```
display:
```

To change the image size and maintain the original aspect ratio, set only one of the *width* or the *height*, not both, and set *aspect* to 1. For example, to grow the image to fill the width of the column, set *width* to -1 and *aspect* to 1. This will maintain the correct aspect ratio as it grows or shrinks. If *width* has a value and *height* is zero, then the height is adjusted for aspect. If *height* has a value and *width* is zero then the width is adjusted for aspect. If both *width* and *height* are set then aspect is not maintained. If *aspect* is zero then aspect is not maintained.

If you want to display a 32-bit partially transparent image (ie the image includes an alpha channel), you will need to set the transparency option to zero in the [options]. This causes the main display to switch off the layered window setting. You can use the formcolor option in [options] to adjust the background color.

To use IMAGE as a banner at the top of the display, use the *global:0* common option.

You can use the various combinations of the *wide, align, column, text* and *display* options to put the image in the left or right column and put text in the other column alongside the image.

For example, put text in the left column and the image in the right column:

IMAGE=file:logo.png, column:1,text:MY COMPANY

Put the image in the left column and text in the right column:

IMAGE=file:logo.png, column:0, display:MY COMPANY

See Also:

Common Options

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

Default Display:
```
display:%9 %10
```

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.

See Also:

[Common Options](#)

LOG

Writes an entry to the log file and/or the Windows Application Event Log. You may include expressions and user variables. There are no return values.

Available Options:

text	the text to write to the log

Default Display:
```
display:%1
```

You must enable writing to the application log by adding the *log* entry and the *user* option to the *log-level* entry in the *[options]* section.

To enable writing to the Windows Application Event Log, add the *eventlog-level* option to the [options] section.

```
 1
 2
 3    [options]
 4    log=desktopinfo.log
 5    log-level=debugonerror user
 6    eventlog-level=user warn
 7
 8    [items]
 9    LOG=interval:60,text:This is my log entry - %var1%
10
11
```

See Also:

[Common Options](#)

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

Default Display:

```
display:%1
```

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,
```

To display all drives except F: and G:

```
exclude:FG,
```

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

```
exclude:AB,
```

To display all drives including drive B: but not drive A:, you need to override the default exclude filter:

```
exclude:A,
```

To display all drives including A: and B:, you need to override the default exclude filter:

```
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,
```

See Also:

[Common Options](#)

MEMORY

Shows physical memory, virtual memory and page file usage.

Return Values:

%1	physical used bytes
%2	physical total bytes
%3	physical used percent
%4	physical free bytes
%5	physical free percent
%6	virtual used bytes
%7	virtual total bytes
%8	virtual used percent
%9	virtual free bytes
%10	virtual free percent
%11	page used bytes

%12	page total bytes
%13	page used percent
%14	page free bytes
%15	page free percent

Default Display:
```
display:%1
```

See Also:

Common Options, PHYSICALRAM, VIRTUALMEMORY, PAGEFILE

MONITOR

Shows metrics about the monitor on which Desktop Info is currently displayed. You can use this to make decisions about display issues such as font sizes etc. There are no options.

Return Values:

%1	monitor number
%2	left position relative to the primary monitor
%3	top position relative to the primary monitor
%4	width in pixels
%5	height in pixels
%6	is it the primary monitor
%7	pixels per inch

Default Display:
```
display:%1
```

See Also:

Common Options

NAV

Creates a group of three navigation buttons consisting of previous, home and next buttons. The context menu is also available when you right click the navigation buttons. This item is ignored in screen saver and preview mode.

Available Options:

font-face	the font for the buttons. The default is Marlett which provides the arrow and square type characters we all know and love.
font-size	the font size for the buttons. The default is 18pt.

color	color of the buttons. Default is #aaaaaa.
button-width	width of the buttons.
caption-prev	caption for the previous button
caption-home	caption for the home button
caption-next	caption for the next button
hint-prev	the mouse-over hint for the previous button
hint-home	the mouse-over hint for the home button
hint-next	the mouse-over hint for the next button
align	alignment for the button group
	-1 = left, 0 = center, 1 =right

To create the standard navigation buttons at the bottom of the page, add the following item at the top of your *[items]*:

```
NAV=global:-1
```

Check out the desktopinfo-advanced.ini file in the sample folder for some alternative nav buttons.

See Also:

[Common Options](#)

NETCONNECTIONS

Shows the current number of network connections.

Return Values:

    %1 number of connections

Default Display:

```
display:%1
```

See Also:

[Common Options](#)

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

Default Display:

```
display:%1
```

See Also:

[Common Options](#)

NETPACKETSRATE

Shows the current ipv4 send and receive rate of network packets. Use the mtu option to convert packets to bytes. I've received reports that this is not so accurate. You should probably use a WMI query instead.

Available Options:

mtu	multiply by this number to get bytes

Return Values:

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

Default Display:
```
display:%1
```
See Also:

[Common Options](#)

NETSTATS

Displays network connections and listeners along with process and traffic statistics in a similar manner to the netstat.exe command line tool or Resource Monitor.

There is a detailed example in the desktopinfo-advanced.ini file.

Available Options:

order	whether the results should be sorted
	0=not sorted (default), 1=sorted
ipv	which ip version to collect
	0=both (default), 4=ipv4, 6=ipv6
filter-module	filter on the module (process) name
	uses the include/exclude syntax (eg: +firefox.exe)
filter-state	filter on the connection state
	a space separated list of state numbers from 1 to 12 (see below)
filter-local	filter on the local address
	uses the include/exclude syntax (eg: +10.0.0.1)
stats	collect traffic statistics
	0=no stats (default), 1=collect stats

Return Values:

state	connection state (numbers from 1 to 12)
	use variables to convert this to human readable.

	See desktopinfo-advanced.ini for an example.
ipv	the ip version of the connection (ipv4,ipv6)
prot	the protocol (TCP or UDP)
laddr	local ip address
	0.0.0.0 = listening on all interfaces
lport	local port
raddr	remote ip address
rport	remote port
laddrf	local address formatted (ip:port)
raddrf	remote address formatted (ip:port)
pid	process id
module	module (process) name
databytesout	number of bytes contained in transmitted data segments
datasegsout	number of data segments sent of positive length
databytesin	number of bytes contained in received data segments
datasegsin	number of data segments received of positive length
segsout	total number of segments sent
segsin	total number of segments received
thrubytesacked	number of bytes sent that have been acknowledged
thrubytesrcvd	number of bytes received that have been acknowledged
databytesoutpersec	rolling average
datasegsoutpersec	rolling average
databytesinpersec	rolling average
datasegsinpersec	rolling average
segsoutpersec	rolling average
segsinpersec	rolling average
thrubytesackedpersec	rolling average
thrubytesrcvdpersec	rolling average

Default Display:

See Also:

https://learn.microsoft.com/en-us/windows/win32/api/iphlpapi/nf-iphlpapi-getextendedtcptable

https://learn.microsoft.com/en-us/windows/win32/api/tcpestats/ns-tcpestats-tcp_estats_data_rod_v0

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

Example:

```
filter-module:+firefox.exe
```

The *filter-state* option is a space separated list of numbers equating to the connection state from 1 to 12. If specified, only the connections that have a state in this list are displayed.

Example: show active connections

```
filter-state:4 5 6 7 8 9 10
```

Example: show listeners

```
filter-state:2
```

Connection States:

> 1 = closed
> 2 = listen
> 3 = syn_sent
> 4 = syn_rcvd
> 5 = estab
> 6 = fin_wait1
> 7 = fin_wait2
> 8 = close_wait
> 9 = closing
> 10 = last_ack
> 11 = time_wait
> 12 = delete_tcb

The *persec* return values are a rolling average of 10 data samples which gives approximately the same results as Resource Monitor.

If you are comparing apples and oranges, a couple of interesting things to note: the GetPerTcpConnectionEStats API call used by Desktop Info seems to be the total stats since the beginning of each process in question while TCPView shows the stats collected since TCPView started.

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 (see below)
	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)

Default Display:

```
display:%1
```

* 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
```

Includes all interfaces that contain the keyword "wireless" and excludes 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 interface and return values %11, %12 and %13 will return the first ipv6 address for the interface.

If you want to display all addresses for the interface, 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.

See Also:

[Common Options](#)

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

Default Display:
```
display:%1
```
See Also:

[Common Options](#)

OSBUILD

Shows the Windows operating system version and build number. On Windows 8.1 and later, the results are unpredictable, use OSBUILD2 instead.

Return Values:

%1	%name%	Windows name
%2	%major%	major version number
%3	%minor%	minor version number
%4	%build%	build number

Default Display:
```
display:%1
```

See Also:

[Common Options](#)

OSBUILD2

This is a much enhanced Windows version to encompass a call to WMI *Win32_OperatingSystem* and the registry *Windows NT\CurrentVersion* and also to accommodate the weirdness of Windows 10/11.

Return Values:

%1	%operating_system%	Windows name from WMI
%2	%architecture%	architecture
%3	%version%	version from WMI
%4	%install_date%	install date
%5	%install_date_short%	install date short formatted
%6	%install_time_short%	install time short formatted
%7	%install_date_year%	install date year
%8	%install_date_month%	install date month
%9	%install_date_day%	install date day
%10	%install_date_hour%	install date hour
%11	%install_date_minute%	install date minute
%12	%install_date_second%	install date second
%13	%install_date_msecond%	install data millsecond
%14	%install_date_utcoffset%	install data utc offset
%15	%major_version%	major version from registry (CurrentVersion)
%16	%minor_version%	minor version from registry (UBR)
%17	%build%	build number from registry (CurrentBuild)
%18	%display_version%	display version from registry (DisplayVersion or ReleaseID)
%19	%windowsdirectory%	Windows directory
%20	%systemdirectory%	system directory
%21	%productname%	product name from registry (same as %1 minus 'Microsoft')

Default Display:
```
display:%1
```

See Also:

[Common Options](#)

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 reduce 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. An alarm 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

Default Display:
```
display:%1
```

See Also:

[Common Options](#)

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

Default Display:
```
display:%1
```

See Also:

PHYSICALDRIVES

Shows information about physical disk drives 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

Default Display:
```
display:%1
```

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

See Also:

PHYSICALRAM

Shows physical memory usage.

Return Values:

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

Default Display:
```
display:%1
```

See Also:

      [Common Options](#)

PROCESSCOUNT

Total number of Windows processes, handles and threads. Some system processes will not provide access to other non-system processes and so the total handles may vary from that which Task Manager reports.

Return Values:

%1	total processes
%2	total handles
%3	total threads

Default Display:
```
display:%1
```

See Also:

      [Common Options](#)

PROCESSMEM

Shows memory and cpu usage for the given process name. If there is more than one instance of the given process name, they are all included in the total. As far as I can tell, page file usage and private usage are the same thing except private usage is not available on Windows 7 and earlier.

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	private usage (commit size)
%6	process handle count
%7	process thread count

Default Display:
>   display:%1

See Also:

> Common Options

PROXY

Shows network proxy server and enabled status.

Return Values:

%1	proxy server
%2	enabled state

Default Display:
>   display:%1

See Also:

> Common Options

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

Default Display:

```
display:%1
```

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 Also:

> Common Options

> 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.

Default Display:

```
display:%1
```

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.

Session State Values

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

See Also:

[Common Options](#)

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

REG

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

Available Options:

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

Return Values:

%1	for a single value
	or the named values for all values

Default Display:

```
display:%1
```

The value option should begin with the root key, followed by the full path to the required key, followed by a backslash and either the required value name or '*'. The root key may be abbreviated to: HKCR, HKCU, HKLM, HKU, HKCC. If you specify a value name, that single value is returned as %1 or the value name. If you specify an asterisk, '*' then all values in the key are returned. The return values may be referenced by number or by name.

Example to return a single value:

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

Example to return all values:

```
value:HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\*
```

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

Do not include *wow6432node* in your key. Use the *wow* option instead. The *wow* option overrides the default behaviour. The default behaviour depends on whether you are running the 32-bit or 64-bit build. The rules are:

```
wow:64 =    KEY_WOW64_64KEY on a 64-bit OS means that the registry
            access, no matter if it's a 32 or 64 bit process, will
            access the 64 bit registry view.

wow:32 =    KEY_WOW64_32KEY on a 64-bit OS means that the registry
            access, no matter if it's a 32 or 64 bit process, will
            access the 32 bit registry view.

Neither of them have any effect on a 32-bit OS. If not specified (the
default), on a 64-bit OS will send registry accesses from 32-bit
processes to the 32 bit registry view and accesses from 64-bit
processes to the 64 bit registry view.
```

For more information on the wow option, go to:

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

REG will return null result if there is an error or the requested data is not available and the error will be written to the log as a DATA entry. This allows you to use the hide-no-result option and the NULL EXPRESSION to eliminate empty results from the display. The following constitutes an error:

- if key is not found
- if value is not found
- if data type is unknown
- if string value is empty
- if binary data type is found

See Also:

> Common Options

REG2

This is the new and improved version of REG. Displays either all values for a given key or a given set of values for all sub-keys. It can only show integers and strings. If value names are specified and none are found then no rows are returned. It can use abbreviated root key names (eg HKLM etc). Wildcards are not allowed.

See a detailed example in desktopinfo-advanced.ini.

Available Options:

key	the registry key to read
wow	overrides default redirection behaviour
	32=force read from the 32 bit registry
	64=force read from the 64 bit registry
host	optional remote host

subkeys	0=return all values for the specified key (default)
	1=return given values (specified in *values* option) for all subkeys, 1 row for each subkey
values	when subkeys=1, this specifies which values to return
	a comma separated list of value names found in the subkeys
	you will need to escape the commas
	ignored if subkeys=0

Return Values:

when subkeys=0, all values for the given key

when subkeys=1, all found values specified in *values*

Default Display:

```
display:%1
```

The key option should begin with the root key, followed by the full path to the required key. The root key may be abbreviated to: HKCR, HKCU, HKLM, HKU, HKCC. Do not include *wow6432node* in your key. Use the *wow* option instead. The *wow* option overrides the default behaviour. The default behaviour depends on whether you are running the 32-bit or 64-bit build on a 64-bit OS. The rules are:

```
wow:64 =    KEY_WOW64_64KEY on a 64-bit OS means that the registry
            access, no matter if it's a 32 or 64 bit process, will
            access the 64 bit registry view.

wow:32 =    KEY_WOW64_32KEY on a 64-bit OS means that the registry
            access, no matter if it's a 32 or 64 bit process, will
            access the 32 bit registry view.

Neither of them have any effect on a 32-bit OS. If not specified (the
default), on a 64-bit OS will send registry accesses from 32-bit
processes to the 32 bit registry view and accesses from 64-bit
processes to the 64 bit registry view.
```

For more information on the wow option, see:

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

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, see:

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

If subkeys=0 and the key is found but is empty then no rows are returned. If subkeys=1, any subkeys that do not contain any specified values are not returned, potentially resulting in no rows returned.

See Also:

> [Common Options](#)

REGEXIST

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

Available Options:

value	the registy key or key\value to be tested
wow	overrides the default redirection behaviour
	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

Default Display:
```
display:%1
```

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

The *wow* option overrides the default behaviour. The default behaviour depends on whether you are running the 32-bit or 64-bit build. For more information on the wow option, go to:

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

See Also:

[Common Options](#)

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

Default Display:

```
display:%1
```

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.
```

See Also:

[Common Options](#)

SERVICEPACK

Shows the Windows service pack edition.

Return Values:

       %1   service pack

Default Display:
```
display:%1
```

See Also:

> [Common Options](#)

SET

Sets the value for the given user variable. See the section on [User Variables](#). It is hidden by default. You can set *hidden:0* for debugging.

Available Options:

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

Return Values:

       %1   the key
       %2   the value

Default Display:
```
display:%2
```

The SET item can be inserted anywhere in the [items] section. IF and SET are the only two items that will be evaluated regardless of the currently displayed page. In other words they are global.

There are two forms of syntax for the SET item. The legacy form is:

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

Or you can use the more familiar form:

```
SET white=#ffffff
```

Where the word prior to the equals sign is the key and anything after the equals sign is the value.

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:

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

Expressions

You can use user variables and expressions in a SET item. See the sine wave example in the *desktopinfo-advanced.ini* file.

See Also:

> [Common Options](#)

SET-SECTION

Loads the contents of a [section] of the ini file into a user variable which can then be used in the *parameters* option of the [CMD](#) item or anywhere else you can find a use for it. Evaluated once on config load.

Available Options:

key	the user variable you want to set
value	the ini [section] to load into the variable

See Also:

      [Common Options](#)

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

Default Display:
```
display:%1
```

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.

See Also:

      [Common Options](#)

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

Default Display:
```
display:%1
```

See Also:

[Common Options](#)

TIMEZONE

Current Windows time zone.

Return Values:

%1	UTC time zone name
%2	UTC time zone bias in minutes
%3	local standard bias in minutes
	used for local time translation when time is standard time
	normally zero
%4	local daylight bias in minutes
	used for local time translation when time is daylight time
%5	current local bias, either %3 or %4 depending if local time
	is currently standard time or daylight time

Default Display:
```
display:%1
```

Where UTC = local time + bias. For the most part when a given time zone is in standard time, that is, not daylight saving time, the local bias is zero. Daylight saving bias is normally -60, that is, the clock is turned forward 60 minutes.

For example, the Sydney time zone UTC bias is -0600 (600 minutes = 10 hours).   During daylight saving we add the local bias of -60, so the local time is -660 (11 hours) relative to UTC.

For more detailed information see:

https://docs.microsoft.com/en-us/windows/win32/api/timezoneapi/ns-timezoneapi-time_zone_information

See Also:

[Common Options](#)

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

Default Display:
```
display:%1
```

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.

See Also:

[Common Options](#)

TOPPROCESSMEM

Shows the processes 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". Microsoft states that on Windows 7 and earlier, Page File Usage is always 0.

Available Options:

counter	which memory counter to use

	ws = Working Set Size	
	pu = Page File Usage (default)	
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	usage bytes

Default Display:

```
display:%1
```

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.

See Also:

[Common Options](#)

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

Default Display:
```
display:%1
```

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.

See Also:

Common Options

UNREADMAIL

Show the number of unread mails.

Return Values:

%1   number of unread mails

Default Display:
```
display:%1
```

See Also:

Common Options

UPTIME

Total time Windows has been running.

Return Values:

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

Default Display:

```
display:%1
```

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

See Also:

[Common Options](#)

USER

Current Windows user account name.

Return Values:

%1	user name

Default Display:
```
display:%1
```
See Also:

[Common Options](#)

VIRTUALDESKTOP

Show the current Windows virtual desktop. If the desktop has a name, that will be shown, otherwise it will show the number. Virtual Desktop names was introduced in Windows 10 20H1. There are no options. See also VIRTUALDESKTOPS.

Return Values:

%1	name (default)
%2	guid

Default Display:
```
display:%1
```

See Also:

[Common Options](#)

VIRTUALDESKTOPS

Show all Windows virtual desktops. If the desktop has a name, that will be shown, otherwise it will show the number. Virtual Desktop names was introduced in Windows 10 20H1. There are no options. See also VIRTUALDESKTOP.

Return Values:

%1	name (default)
%2	guid

Default Display:
```
display:%1
```

See Also:

[Common Options](#)

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

Default Display:
```
display:%1
```

See Also:

[Common Options](#)

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.

Default Display:

```
display:%rowcount%
```

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 it's 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 it's 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/

See Also:

Common Options

Appendices

Appendix 1: 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
```

Display Adapter

```
WMI=text:Display Controller,namespace:root\
cimv2,query:Win32_VideoController,display:%Caption%| Chipset:
%VideoProcessor%| Video Memory: %AdapterRAM%[1.0b]B| Driver Version:
%DriverVersion%| Driver Date: %DriverDate_year%-%DriverDate_month%-
%DriverDate_day%
```

Display Resolution

```
WMI=text:Display,namespace:root\
cimv2,query:Win32_VideoController,display:%VideoModeDescription%
```

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://openhardwaremonitor.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:%Workgroup%
```

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
```

Appendix 2: 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
```

Appendix 3: 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

OpenSSL

www.openssl.org

OpenSSL is a robust, commercial-grade, and full-featured toolkit for the Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols.

Appendix 4: 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.

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

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.

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 problemwhen 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 inimonitortime 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.

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.

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.

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.

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 "InetNtopw not found ws2_32.dll" bug running under Windows XP.

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.

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.

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 v.1.16.0

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.

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_DEVTYPE_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.

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.

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.

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.

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.

Version v2.11.0

June 2021

Change 1: Fixed bug in CMD when the result is exactly 200 bytes.

Change 2: In CMD, don't trim the result if *read-as-csv* is specified.

Change 3: The FILE item has been converted to a regular item. It kind of always was but there was some old code in there that ran in a thread that is unnecessary. I haven't included the "first line of text" option. Does anyone use that? The return values have changed, check your config.

Change 4: Added alternative syntax to the SET item. You can now use the more conventional *SET key=value* syntax.

Change 5: Added user variables to *interval* and *background-interval* options. These are only parsed once.

Change 6: Added Wow64DisableWow64FsRedirection calls to the CONTROL and CMD items to disable file system redirection from System32 to SysWOW64 and the like.

Change 7: Added the *offset* option to the DATETIME item to display times in other time zones.

Change 8: Added a buffer to the ini file reader to reduce disk operations and allow the following.

Change 9: Added the INCLUDE key word to the ini file.

Version v3.0.0

July 2021

I have finally migrated the project to the latest Embarcadero Delphi compiler. This is a leap of technology of 16 years so there's a metric tonne of changes to make it work. The upside is we now have 64-bit and 32-bit builds with access to the latest tools and language technology. The downside is because there's a metric tonne of changes, I'm expecting broken stuff so please report your findings to the forum including the build number, 32-bit/64-bit and particularly the Windows version. But before you report, make sure it's not because of some config change outlined below.

The migration made it impossible to support Windows 2000. Desktop Info now supports Windows XP and up. If you need support for W2K then you should use version 2. I haven't found any other issues specific to a version of Windows.

The zip contains two new executables: DesktopInfo32.exe and DesktopInfo64.exe. These are the new 32-bit and 64-bit versions of Desktop Info. The old DesktopInfo64.exe external collector tool is gone.

Your old ini file should work for the most part, though it will require some modification as outlined in the changes below. I haven't tested every single configuration possibility.

Change 1: All the charts except for the horizontal bar chart have been thrown out and replaced with brand new charts. For the most part they work the same as the previous charts. Two new chart options have been added: points and border.

Change 2: Added the new function rnd() to the mathematical functions (great for testing charts!).

Change 3: Replaced the navigation button component.

Change 4: Replaced the png image component.

Change 5: Completely rewritten the CPUUSAGE and CPUCOUNT code. Can anyone confirm the maximum number of cores this can handle?

Change 6: Added *filter* option to the CPUUSAGE item.

Change 7: Completely rewritten TOPPROCESSMEM. The external tool is no longer used. The *counter* option now simply controls which counter is used to determine the top processes.

Change 8: Completely rewritten PROCESSMEM. The external tool is no longer used.

Change 9: Completely rewritten PROCESSCOUNT. The external tool is no longer used.

Change 10: Removed the *desktopinfo64* option in the [options] section. This external tool is no longer required.

Change 11: EVENTLOG somewhat rewritten to handle new strings and pointers and things. Added additional event data return values.

Change 12: RDS and RDSSESSIONS somewhat rewritten to handle new strings and pointers and things.

Change 13: The remote monitor data server and it's associated item DTI have been rewritten for new XML code handling and the new internet tools.

Change 14: The REG and REGEXIST items now no longer try to override the wow redirection by default. The behaviour is as you would expect from a 32-bit or 64-bit application. The *wow* option remains if you wish to override the standard behaviour.

Change 15: Added the IMAGE item. This makes the banner logo redundant and so will eventually be removed.

Change 16: The *active* item option has now been removed. You should use the PAGE keyword instead to set item pages.

Change 17: The [pages] section in the ini file has now been removed. You should use the PAGE and PAGE-TITLE keywords instead to define pages.

Change 18: Fixed build number in FILE item.

Change 19: Removed *bar-colors* option. The bar colors are set within the chart option.

Change 20: Fixed OSBUILD to show Windows 10 correctly. Added more return values.

Change 21: Modified FILE2TEXT to return %1 and %2 return values. This way you can use the display template in the normal way to display the results.

Change 22: Modified FILECONTENTS to return %1 return value. This way you can use the display template in the normal way to display the results.

Change 23: Added additional return values to TIMEZONE.

Change 24: Updated DesktopInfo-Technician.bat to detect the architecture and run the correct executable.

Change 25: Completely rewritten NETWORKINTERFACES.

Change 26: Added locale decimal separator to the expression parser. If you want to add your own floating point values to an expression and you're in a locale that uses a comma as a decimal separator, then remember to escape the comma with a backslash. Eg: {{%1 * 1\,25}}

Change 27: Changed the logging so that *debugonerror* will write a stack to the log when an error is raised. This shows events leading up to the error. This is exactly the opposite of how it used to work where it would write 10 entries after the error. Generally improved the logging across the board.

Change 28: Added the *top* and *tail* options to the FILECONTENTS item.

Change 29: There are hundreds of small changes to make the new strings work.

Version v3.1.0

August 2021

The major overhaul continues, nothing is safe! Still converting the code to the new tools, strings and high precision numbers and boldly going where no-one has gone for some time.

If you find your context menu options have gone missing, copy and paste the *[text] section* from one of the sample ini files.

Change 1: HTTPGET fixed to correctly return UTF-8 content. Added http connection/read timeout.

Change 2: Fixed the broken screen saver. Even got the preview working properly.

Change 3: Converted the DTI item to new internet code. Now defaults to port 80 if not given. Added the *name* field to the DTI item XML so that you can now reference the WMI columns by name. Other fine tuning.

Change 4: Added some CORETEMP fields.

Change 5: Fixed LOGICALDRIVES *diskio* floating point precision. Also, when the performance data is not collected or not available, set the data to zero to avoid expression parser errors.

Change 6: Added environment expansion to the log file name so you can add *%appdata%* to the log file path. This allows you to install it in Program Files without it needing to write the log in that path. See the discussion from a user on secure installation.

Change 7: No longer recreating the data collector thread or the remote server thread when the configuration is reloaded.

Change 8: I've implemented a simple way for developers to provide data to Desktop Info by writing external collectors. Use the COLLECTOR item to display external collector data within Desktop Info.

Change 9: Export to RTF is now correctly encoding Unicode characters.

Change 10: Removed the limitation on the number of pages.

Change 11: Changed the way item ids work. The old *lid* option is gone. The existing *id* option is now automatically assigned a unique value for every item. For example, *wmi, wmi1, wmi2, wmi3* etc. You may override this by adding the *id* option to the item, it is now up to you to ensure that the *id* is unique. This id is used in a number of places such as DTI.

Change 12: I have removed the language overlays. It was all rather awkward and not really useful. If you want to translate Desktop Info just modify the configuration file directly. Please send me your translated configurations so that we can share with others.

Change 13: Added automatic scaling for high dpi devices and Windows scaling. Added the *scale* option to *[options]* to enable or disable automatic scaling. It is on by default.

Change 14: Added log file to context menu.

Change 15: The context menu will now split into columns when there are too many entries.

Change 16: I've changed the way the configurable menu options work. If a given entry exists in the *[text]* section then the menu item will appear. Therefore to remove the menu item, remove the *[text]* entry by deleting it or commenting it out. Also, for the log menu item to appear, both the log entry and it's *[text]* entry must be configured. If there are no templates found, the export menu is not displayed. If your context menu is missing entries, your ini file is probably missing the [text] entries, just grab them from the *sample-config\desktopinfo-sample.ini* file.

Change 17: FONT-FACE and FONT-SIZE may now include user variables. There are now five KEY WORDS that respond to user variables.

Change 18: Adjusted FONT-SIZE so that when it's scaled up or down using percents, it is always relative to the last absolute value given. This way when FONT-SIZE=100% is given, it will correctly return to the base value. Previously, scaling using percents would accumulate.

Change 19: Removed DISPLAYCONTROLLER item. Look at *desktopinfo-advanced.ini* for a better alternative.

Change 20: Fine tuned the remote monitor server settings. See Remote Monitoring for details.

Change 21: Added the BEGIN-ONLOAD / END-ONLOAD key words for evaluating items during configuration load up and added WIDTH key word.

Change 22: Added the /o command line option.

Change 23: Added new functionality to the IF item. It can now set a user variable depending on the comparison outcome.

Change 24: Added the LOG item. Added the *user* option to both the application logging and the Windows Application Event Log so that the new LOG item can write to either or both.

Change 25: Added the MONITOR item to show metrics about the current monitor.

Change 26: Added user variables and expression parsing to the SET item. Removed the recursion so that a SET item variable can reference itself (eg. set xx={{%xx%+1}}).

Version v3.2.0

September 2021

Important: Make sure to update your expressions/function calls to relocate your display format definition as outlined below.

My TO DO list is nearly exhausted so the rampage will begin to slow down now.

Change 1: In the process of cleaning up the data processing code, I've removed the display format definition from inside expressions and functions and placed it after the closing curly braces so that it more closely follows the convention used elsewhere. This has been bugging me for a while and I decided that while I'm on a rampage, that's as good a time as any. So now the sole function of anything inside the double curly braces is to calculate a value, either by explicit expression or by calling a function. Once this is done and the expression or function has been fully evaluated, any following display format definition will format it for display. See Functions Section and Expressions and Functions. This should make things easier to fathom. *** You will need to change your configuration file to move the display format definitions out of the expressions.

Change 2: Completed the SSL implementation on the remote monitor, DTI and HTTPGET. I've had some questions about the extra DLL files in the distribution zip. These are the OpenSSL files used to provide encryption on secure connections. See www.openssl.org. They integrate nicely with the Indy components used to provide network connections.

Change 3: Changed the behaviour of Desktop Info when the *transparency option* is set to zero. Now, the main display form becomes a non-layered window. This allows 32-bit partially transparent (ie with an alpha channel) images to display correctly, albeit on a plain colored background. As far as I can tell it is not possible to display such images correctly on a layered/transparent window.

Change 4: The collector thread will try to throttle itself when there is nothing else going on and it looks like Desktop Info is the highest cpu user. This requires a TOPPROCESSCPU item to be active, visible or hidden.

Change 5: Overhauled the EVENTLOG item. Added the *maxrows* option to allow more than one recent event. Added the *filter* option to provide filtering on the source field. Now supports multi-byte.

Change 6: Added the secret command line option.

Change 7: Added the *topmost* option.

Change 8: Added a new alarms option to the items. This new option is much more powerful than the thresholds. Not only does it allow a variety of ways to compare the data for potential alarms but it also allows you to compare the current data against the previous dataset so it can detect changes in data. It will also compare any two values and/or expressions whether or not they come from the current dataset. Thresholds are deprecated.

Change 9: Added a new ALARMS item to display a list of recent alarms generated by the aforementioned *alarms* option in each item.

Change 10: A lot more work on scaling, hopefully I've fixed the various issues. Changing the Windows magnification on the fly and dragging between monitors of differing dpi still don't work correctly. The *scale* option is removed.

Change 11: Added the Not Null Expression option to the item display template. I couldn't think of a better name for it.

Change 12: Added the 's' format to help in creating a table. See the *sample-config\desktopinfo-linux.ini* configuration file for an example.

Change 13: Added VIRTUALDESKTOP to show the current Windows virtual desktop and VIRTUALDESKTOPS to show all Windows virtual desktops.

Version v3.2.1

September 2021

Change 1: Added *maxrows* option to the COLLECTOR and DTI items.

Change 2: Fixed not null expression on HTTPGET and NETWORKINTERFACES.

Change 3: Fixed lease lifetime on NETWORKINTERFACES.

Change 4: Added return values to CPU item.

Change 5: Allow ini file position options to be used in /f mode.

Change 6: Changed GetUrlContent error to warning. Used by HTTPGET and DTI items.

Change 7: Changed TopProcessCpu percent value to floating point. You might have to add a display format definition to this return value. Eg: %3[1.1f]%

Change 8: Resolved a thread synchronisation issue.

Change 9: Fixed control offset bug on high dpi monitors.

Version v3.2.2

September 2021

Change 1: All charts except the horizontal bar chart can now do automatic maximum values. Which is to say if you do not specify the *max* option, the maximum value of the chart will be determined by the highest value currently on the screen. The vertical scale will adjust itself appropriately.

Change 2: Fixed bug where the IF item would sometimes cause the END-ONLOAD to not evaluate.

Change 3: Fixed another drag/not drag ini change position bug.

Change 4: Fixed bug where the MONITOR item would crash if evaluated inside the ONLOAD block.

Change 5: I may or may not have fixed the high dpi font quality issue. Windows is giving me mixed messages on this one.

Version v3.2.3

October 2021

Change 1: Fixed control right align.

Change 2: REG will now return null result instead of an error string if there is an error or the requested data is not available and the error will be written to the log as a DATA entry. This allows you to use the hide-no-result option and the NULL EXPRESSION to eliminate empty results from the display.

Change 3: More work on scaling controls.

Change 4: I've definitely got the font quality this time! Added the new quality option, Clear Type Natural.

Change 5: Fixed issue with no results in ALARMS.

Change 6: Added the OSBUILD2 item to handle the weirdness of Windows 10/11.

Version v3.2.4

October 2021

Change 1: Fixed OSBUILD2 for null expressions.

Change 2: Added registry product name to OSBUILD2.

Change 3: More work on scaling. Added *auto-scale* option. Scaling is not an option for Windows below version 10. Scaling is on by default for Windows 10 and later. It can be disabled by setting *auto-scale=0*. If the Desktop Info scaling is not working properly on your device, you should disable it and use the Windows compatibility settings.

Change 4: Fixed a few data titles so that the return values can be referenced by name rather than number. These titles have always been there but undocumented (they show up in the csv export). Eventually I'll document them all, meanwhile if you want to know any specific ones, drop me a message on the forums.

Change 5: The REG item can now return all values from a given key.

Change 6: Tested on Windows 11 Pro 22000.258.

Change 7: Fixed a bug that was preventing it from running on some Windows 7 installations.

Desktop Info - Manual

Version 3.3.0

November 2021

Change 1: Adjusted OSBUILD2 DisplayVersion/ReleaseId to return null if neither is found.

Change 2: Added *onload* common item option. This allows an individual item to be evaluated during configuration load and also saved to the active config in the normal way as opposed to items in the BEGIN-ONLOAD block which are **not** saved and hence never evaluated again.

Change 3: Fixed up some little oddities in the BEGIN-ONLOAD logic.

Change 4: I've had a number of enquiries regarding setting item colours based on IF item results. It took me some time to understand it myself which is bad. The complication comes down to the colours being static, they're set once at configuration load. So I've made the item colours dynamic so you can now use variables and expressions for the *color* and *text-color* item options and these will be re-evaluated each time the item is evaluated allowing the colours to change on the fly as the variables change.

Change 5: In the preconfigure option, I've added code to remove the existing code certificate before it embeds your ini file. Unfortunately, if you try to resign the executable it will trash the embedded ini. I have to work out how to update the exe header so it includes the embedded ini.

Change 6: Adjusted control visibility so it works correctly following an IF item.

Version 3.3.1

December 2021

Change 1: I've rewritten CPUUSAGE to load the local language version of the performance counters and to be more robust. Also, it now responds correctly to a reload.

Change 2: Some minor defensive code additions.

Version 3.4.0

February 2022

Change 1: Added <dti-filter> option to the export tool.

Change 2: Added */silent* argument to the preconfigure tool.

Change 3: Fixed bug in *auto-scale* during config load (default is now off).

Change 4: Strip %chart% from export.

Version 3.5.0

March 2022

Change 1: Updated VIRTUALDESKTOP for Windows 11. Sometimes, if the user is not using the virtual desktops, Windows will not record the current virtual desktop. In this case Desktop Info can only assume you're using the first, always active one.

Change 2: Added SET-SECTION item. This loads a [section] of the ini file into a user variable which can then be used within the *parameters* option of the CMD item. This way you can embed your external CMD scripts into the ini file. Makes things a little more portable.

Change 3: The *parameters* option in the CMD item can now accept user variables.

Change 4: Removed the *sample-scripts* folder. These scripts now occupy their own sections in the *sample-config\ desktopinfo-advanced.ini* file.

Change 5: Removed the default text from the IMAGE item.

Change 6: Reworked the *display* output so that the *wide* option works properly in all (most?) cases.

Change 7: Added Neg() function to expressions. This doesn't force a value to be negative, it merely toggles it (x = -x).

Change 8: Pretty much rewrote the charts: now correctly displays positive and negative values, make a mirror series using the Neg() function, specify the max and min values or let them be automatically calculated, displays the zero line calculated from the current max and min values.

Version 3.6.0

July 2022

Change 1: Added the *share-data* option to the *[options]* section of the ini file. This enables local data sharing and acts like an external collector. A secondary instance of DTI can then retrieve this data using the *COLLECTOR* item for display. This provides a mechanism for creating multiple DTI displays, say on secondary monitors, without incurring the overhead of additional data collection.

Change 2: When retrieving data from a shared memory area, the required access is now READ instead of ALL_ACCESS. Added error logging.

Change 3: Removed deprecated thresholds. Use alarms instead.

Version 3.7.0

August 2022

Reminder: The following options are deprecated and will be removed in a future release:

- In the *[options]* section: nav, nav-align, nav-color, logo, logo-width, logo-height, logo-align
- In the *[text]* section: nav-previous, nav-home, nav-next

For a banner logo, use the *IMAGE* item. To place the logo at the top of every page, add the *global:0* option. For standard navigation at the bottom of each page, use the *NAV* item with the *global:-1* option.

Change 1: Added the *eval* option to the *IF* item. This sets the number of following items that will be evaluated if the *IF* item returns *TRUE* and hence the items that will be skipped if it is FALSE.

Change 2: Updated the *onload* common option so that it works correctly with the *eval* option in the *IF* item.

Change 3: Added user variables to the *font-size* option.

Change 4: Added the *global* common option so that you can define an item once and show it on every page. Useful for banners, graphics, messaging, navigation etc.

Change 5: Added the NAV item. This is basically a replica of, and deprecates, the old *nav* entries in the *[options]* section and *[text]* section. Eventually I'll enhance this to make it more flexible. A direct replacement of the old navigation would be: NAV=global:-1. This creates navigation buttons at the bottom of each page.

Change 6: Added the *no-wait* common option. This allows long running items to be evaluated in their own thread without blocking the main collector thread. Keeps the display ticking over.

Version 3.8.0

October 2022

Remember to change over to the new NAV item and IMAGE item for your logo.

Change 1: Fixed the line chart bug where it displays the odd line at the top of the display.

Change 2: Added user variables to HTTPGET source option.

Change 3: Fixed chart overflow bug. Charts can now handle huge 64 bit values. Code added to hopefully avoid crashing on vastly monumentally insanely big values.

Change 4: Fixed NETPACKETSRATE overflow bug. Set rate types to 64 bit.

Change 5: Added a bunch of options to the NAV item.

Change 6: Added *width* and *height* options to CONTROL items and in the process rewrote the layout code for controls yet again. Width and height can be set for both hyperlinks and buttons. In both cases, if not specified, they are calculated from the displayed text. You still need to set the *left* option. The height of the display row is set by the height of the biggest control. Both controls will now word break where the control is not wide enough.

Change 7: Added ALIGN and TEXT-ALIGN key words so they can be used throughout the ini to set global values.

Change 8: Added the center option to both the ALIGN/TEXT-ALIGN key words and local item options. Doesn't yet work on controls. This necessitated a rewrite of the alignment code and in the process fixed a couple of layout issues.

Change 9: Disconnected the font-rtl option from right align options. They both work independently, you can set one or the other or both.

Version 3.8.1

November 2022

Change 1: There was an undocumented common option called multirow which is now documented.

Change 2: Fixed the arbitrary buffer size and resulting crash when loading the configuration file.

Change 3: During configuration load up, when the IF item evaluates to FALSE, it will make the following "eval" number of items inactive rather than not load them at all. This allows those items to be activated again if the conditions change.

Change 4: The IF item now correctly activates and deactivates CONTROL items.

Version 3.9.0

December 2022

This release has a high number of code changes for relatively few entries in the release notes which makes for a greater risk of breakage. Let me know in the forums (after you've read the changes) if you believe something is broken.

The old, deprecated NAV buttons in the *[options]* are gone. You must change over to the new NAV item now. Just add "*NAV=global:-1*" to the top of the *[items]*.

Change 1: Added regular expressions as a common option. This helps you to massage the raw data before it is inserted into a display row or assigned to a variable. Just don't ask me to write your regular expressions.

Change 2: Added the NETSTATS item to list network connections ala the netstat cmd tool. I'm still working on traffic statistics, that's proving elusive.

Change 3: In the REG item, the *value* option can now handle the usual root key abbreviations: HKCR, HKCU, HKLM, HKU, HKCC.

Change 4: Added the REG2 item. This was written specifically so I can get the Windows uninstall list from the registry and is overall a better version of REG which may eventually be deprecated. The desktopinfo-advanced.ini file has a comprehensive uninstall display now.

Change 5: Added the WIDE key word. COMMENT is the only item that is natively WIDE.

Change 6: Fixed an issue with incorrect chart background colours in some circumstances.

Change 7: Fixed an issue with hyperlink background colours in some circumstances.

Change 8: The zip file now includes a special "W" build. This is essentially a much improved version of the "/f" command line switch but now has it's own dedicated executable. Consequently the "/f" command line switch has been removed and the *desktopinfo-technician.bat* batch is no longer required. I'll work on making this build more functional over the next little while.

Change 9: The old, deprecated navigation buttons are gone. You should have changed over to the new navigation by now. To do this just add "*NAV=global:-1*" to the top of the *[items]* section. The *nav* entries in the *[text]* section are now used by the toolbar buttons in the "W" mode build.

Version 3.10.0

January 2023

Change 1: More work and additions to NETSTATS. Now shows traffic stats for TCP connections. Still researching UDP traffic. The return values have changed.

Change 2: Added option to start W mode to the context menu. It's caption text in the *[text]* section is *menu-wmode*.

Change 3: Added support for user variables to IMAGE *file* option during load-up.

Change 4: Added the *key* and *value* values to the SET return values for debugging. Set hidden:0.

Change 5: Expanded display height for very long displays. In 'W' mode this is now possible.

Change 6: Fixed oversight with key words in the BEGIN-ONLOAD section.

Change 7: Added the *[options-w]* section to the ini file. This provides an alternative set of options for the W mode, DesktopInfoW.exe. You can add *toolbar=0* to disable the top toolbar in W mode.

Change 8: W mode process priority is set to normal.

Version 3.10.1

March 2023

Change 1: IMAGE now evaluates every cycle according to it's interval setting so you can change the image on the fly. The down side is it will now attempt to load the image every time it evaluates. Set the interval appropriately. The default interval is 0, evaluate once.

Change 2: Added the MEMORY item which is a single call encompassing PHYSICALRAM, VIRTUALMEMORY and PAGEFILE.

Change 3: Added the centerv and centerh options to the [options-w] section. These are used initially to position the form and then it behaves normally.

Version 3.10.2

May 2023

Change 1: The IMAGE item is now playing better. I've added the *column* option to put the image in the left or right column. Along with the *wide*, *width*, *align*, *offset* and other options, we have better control of the layout and also the ability to put text alongside the image. For example, put the image in the left column and use *display* to put text in the right column. Or put the image in the right column and use *text* to put text in the left column.

Change 2: Fixed application crash in *auto-home* option.

Change 3: Fixed unknown file extension 'gif' on the IMAGE item. No animation yet.

Version 3.11.0

October 2023

Change 1: Added FORMCOLOR, TRANSPARENCY, DISPLAY-OFFSETX, DISPLAY-OFFSETY, WIDTH, HEIGHT to both the [options] section and the [items] section. This allows you to make decisions about the display layout during load. See the Options section and keywords for an example.

Change 2: Fixed global IMAGE bug.

Change 3: Fixed IMAGE bug where only one line of text can appear next to the image.

Change 4: Fixed the hyperlink alignment. 0=left 1=right 2=center

Change 5: Removed the deprecated banner logo.