



SBSH PocketWeather Custom View Syntax

For PocketWeather 2.3.0



By: Adrian Oliver
Ronald A. Bernstein, M.D.

Last Updated: 1 July 2009

Downloads and additional support can be found at our site

<http://www.sbsh.net>



Table of Contents

Table of Contents.....	2
Overview.....	5
Files and Folders.....	6
Strings & StringIDs.....	6
Standard Folder String IDs – Translation Table 1.....	6
Standard File String IDs – Translation Table 2.....	7
Standard Text IDs – Translation Table 3.....	9
File Content & Syntax.....	11
Comments.....	12
Script Development Aids - Debugging & Indentation.....	13
Debugging Commands – Table 4.....	13
Custom Layout Script - A Simple Example.....	14
D-Pad Navigation & Gestures.....	15
Objects.....	16
OBJECT Commands – Table 5.....	16
Object Positioning and Dimensions.....	17
Alignment.....	18
Horizontal/Vertical Alignment Attributes – Table 6.....	18
Font Specification.....	19
Font Attributes – Table 7.....	19
Colors.....	20
Color Today/Home Screen Special Cases.....	20
Default Colors.....	20
Rotation.....	20
Heading Commands.....	21
Heading Commands – Table 8.....	21
DRAWTABS.....	25
DrawTab Commands – Table 9.....	26
Image Objects.....	28
ImageTypes – Table 10.....	28
Icon Sizes.....	29
Day/Night Icon	30
Optional Rotation Specification.....	31
Standard Image IDS – Table 11.....	32
Draw Objects.....	35
Alpha-Shading.....	36
Draw Objects With Variable Size.....	37
Attributes – Table 12.....	37



Draw Line With Variable Angle.....	38
Supported Variables – Table 13.....	39
Historical Variables.....	42
Historical Variables – Table 14.....	42
Graph Commands – Table 15.....	43
Display Attributes – Table 16.....	43
Text Object.....	45
Date/Time Objects.....	45
Time and Date Variables – Table 17.....	46
General Variables.....	48
General Variables – Table 18.....	48
Current Conditions Weather Template Related Variables.....	49
Current Condition Variables – Table 19.....	49
Forecast Weather Template Related Variables.....	50
Forecast Variables – Table 20.....	50
Historical Weather Template Related Variables.....	51
Historical Variables – Table 21.....	51
Historical Variables Within Weather Loops – Table 22.....	51
Weather Alert Template Related Variables.....	52
Weather Alert Variables – Table 23.....	52
Earthquake Variables.....	52
Earthquake Variables – Table 24.....	52
Debugging Variables.....	53
Debugging Variables – Table 25.....	53
String IDS.....	53
Location Command Object.....	54
Location Command – Table 26.....	54
Time Object.....	55
Time Command Attributes – Table 27.....	55
Sub Views Object.....	56
Touch Object.....	57
Touch Area Ordering.....	58
WeatherConsole Command Line Options.....	59
WeatherConsole Command Line Options – Table 28.....	59
Conditional Statements.....	60
Conditional Statements - Conditions.....	61
Conditional Statements – Table 29.....	61
Conditional Operators.....	63
Conditional Operators – Table 30.....	63
Conditional Statements – “NOT”.....	64
Conditional NOT Examples – Table 31.....	64



CUSTOM Positioning.....	65
Selecting Weather Data Context.....	66
Forecast Data.....	66
Forecast For Specific Day.....	68
Forecast From Today.....	69
Forecast From Tomorrow.....	69
Hourly Forecast.....	70
Selecting Historical Data.....	71
Current Conditions.....	72



Overview

There are several ways to customize the display of data and images in PocketWeather. In addition to a host of Options, one can use a custom layout to create a custom view. The PocketWeather Today item (plugin) screen view and WeatherConsole both support views which can be customized, allowing you to configure PocketWeather to display weather data and images in pretty much any layout and style you wish.

When first installing PocketWeather, you are provided with a number of different layouts which can be selected from the MainMenu. These default layouts are organized into folders on your device, customized for either the Today screen view or WeatherConsole; however, there is nothing preventing you from selecting a Today screen view for WeatherConsole, or a WeatherConsole view for the Today screen -- they are interchangeable.

Obviously a Today screen view will not use the full display area available to WeatherConsole, while a WeatherConsole view may appear squashed or cropped into the limited Today screen display area and may require reformatting.

If you cannot find a layout that suits your needs from the default layouts or from the constantly growing supply of layouts that users are producing, then this document will help you to either tweak an existing layout or generate your own new custom view.



Files and Folders

The files that PocketWeather uses for CustomViews must end with the ".PWC" file extension so that the layout name will appear on the menu. These are simple plain-text files which can be edited with your favorite text editor, such as Notepad. Although ANSI text encoding is allowed, it is strongly recommended that UNICODE (big endian) encoding be used.

To install a new CustomView file onto your device, simply copy the ".PWC" file into the "Skins" folder on your device.

To cause this new CustomView file to be displayed, use the PocketWeather main menu and select Layouts (may be under TodayView).

To organise the CustomView files, it is recommended that new folders (or sub-folders) be added under the "Skins" folder; e.g., you could add a folder called "mine" under "skins".

The skin folders and files will appear in PocketWeather in terms of hierarchy and names as they are declared in the device's folder structure. This however also means that a folder called say "My Today Layouts" will always appear as "My Today Layouts" in all locales and languages. The same applies to file names.

Strings & StringIDs

Standard folder names, standard files names and standard text strings (as well as other miscellaneous text strings), are automatically translated into your specified language by using numeric values (StringIDs) which map to PocketWeather's Translation Strings. The term Standard is used throughout this manual to mean internally reserved. Below are the StringIDs and their English Language (US) translation strings.

If you would like additional translation string IDs to be added to PocketWeather's translation table, please send any suggestions to adrian.oliver@sbsb.net.

Standard Folder String IDs – Translation Table 1

ID	English (US) Language String
47000	"Today Extras"
47001	"Today Plugin"
47002	"WeatherConsole"
47003	"Non Tabbed"

**Standard File String IDs – Translation Table 2**

ID	English (US) Language String
47100	"Detailed"
47101	"Summary"
47102	"Classic"
47103	"Classic Forecast"
47104	"Classic Navigator"
47105	"Compact"
47106	"Compact Forecast"
47107	"Single Line"
47108	"MultiLine 2"
47109	"MultiLine 3"
47110	"MultiLine 4"
47111	"Bold Horizontal"
47112	"Bold Horizontal Large Icons"
47113	"Bold Horizontal Small Icons"
47114	"Bottom Horizontal"
47115	"Classic with Analogue Clock"
47116	"Classic with Digital Clock"
47117	"Classic No Images"
47118	"Classic No Images with Units"
47119	"Classic with Units"
47120	"Compact Retro"
47121	"Compact Vertical"
47122	"Compact Vertical with Name"
47123	"DOS"
47124	"Retro"
47125	"Vertical"
47126	"Vertical Retro"
47127	"Classic No Images" (see note below)
47128	"Classic No Images with Units" (see note below)
47129	"Classic" (see note below)
47130	"Classic with Units" (see note below)



ID	English (US) Language String
47131	"Compact" (see note below)
47132	"Compact Retro" (see note below)
47133	"Moon"
47134	"Alerts"
47135	"METAR"
47136	"MultiLine 5"
47137	"MultiLine 6"
47138	"Historical"
47139	"Overview"
47140	"Multiple Clocks"

**** Note** - some strings may appear duplicates in the above table. This IS deliberate to overcome the CAB file installation issue where each filename has to be unique, even when in different folders.

Other standard strings can be selected and are automatically translated into the configured language.

**Standard Text IDs – Translation Table 3**

ID	English (US) Language String
30001	"Sun"
30002	"Moon"
30004	"Precipitation"
30005	"Sunrise"
30006	"Zenith"
30007	"Sunset"
30008	"Actual"
30009	"Apparent"
30010	"Wind"
30011	"Visibility"
30012	"Humidity"
30014	"Current Conditions"
30015	"Today's forecast"
30016	"Day"
30017	"Night"
30018	"Recent Earthquakes"
30019	"Images"
30027	"RH"
30029	"Pptn"
30031	"Pressure"
30032	"Maximum"
30033	"Minimum"
30042	"observed at"
30043	"downloaded"
30048	"Today"
30049	"Dew Point"
30061	"Time"
30066	"Temperature"
30077	"at"
30078	"Gust"



ID	English (US) Language String
30079	"Gusting"
30085	"No active alerts available"
30100	"Max"
30101	"Min"
30103	"Reset"
30133	"Downloaded at"
30308	"Apply"
30309	"Exit"
30325	"Close"
30338	"Dismiss"
31667	"Hourly Forecast"
31680	"Next"
31681	"Previous"
31682	"Start"
31683	"Stop"



File Content & Syntax

The CustomView text file contain a series of instructions or commands that tell PocketWeather where to place the various elements that make up the CustomView display.

To be functional, each command must be entered in a particular way: this command entry structure is known as the command's "Syntax". Failure to follow the syntax precisely way result in unexpected results.

The Command Name is always entered after a semicolon (";") in UPPERCASE capital letters. Lower case or mixed case commands may be ignored or produce unexpected results. The Command Name spelling must be correct.

Immediately following the command name on the following line(s) are the various command specific attributes or parameters, e.g. position, font, color, etc. These attributes will typically span several consecutive lines; there must not be any empty lines between these attributes.

Not all commands require attributes. In a few cases there are shortcuts where command names or attributes can be abbreviated.

The name of a parameter is enclosed in angle brackets "< >". *Italicized* items need to be replaced with specific values. Items enclosed in square brackets "[]" are optional; the command will work in its basic form without entering the information contained inside the brackets. When items are separated by a vertical bar "|", it means that you can enter one of the separated items, but not both.

Example:

```
ON | OFF
```

Detailed explanations of each command will be presented and may include an example and image.

Syntax:

```
; COMMAND  
<AnyString>  
<BoundingRectangle> [, <Alignment>]  
[<FontSpecification>]
```

Example:

```
;TEXT  
Welcome to PocketWeather Scripting  
0, 10, W, 15, C  
Tahoma, 10, BOLD, 0-0-0
```

The command name and its attribute(s) combine to form an Object, e.g., the TEXT object above.

A layout script will consist of multiple objects. Each object must be separated by at least one blank/empty line.

By adding commands and objects of various types with specific attributes, you can construct your own CustomView layout.



Comments

A comment is defined as having either a "-" before and/or after the first line of the object.

Example:

- This is a comment

-This is also a comment

So is this a comment-

;DRAW-

RECTANGLE PENCOLOR-0-0-0 BRUSHCOLOR-250-250-240 2
0%,14,100%,64

The above object is completely commented out because the ";DRAW" has a "-" afterwards, resulting in the complete object being disabled (i.e. the next two lines).

For software developers, "//" is also a comment.

Example:

// this is a comment



Script Development Aids - Debugging & Indentation

PocketWeather provides a number of features to aid the script developer. These include the RELOAD command and several DEBUG commands described later as well as an indentation feature. You can also "hide" Current Conditions to test conditional loops.

Leading white space and tabs are ignored. This means that scripting can be suitably indented to help code readability and delineate logical constructs.

In addition, you may place a "title" above an Object for further documentation and script clarity.

Example:

```
- This is a valid title for an executable DRAW object
;DRAW
RECTANGLE PENCOLOR-0-0-0 BRUSHCOLOR-250-250-240
20%,14,100%,64
```

PocketWeather provides a number of debugging options to allow the script developer to see where each object is drawn, and how the script is being executed.

Debugging Commands – Table 4

Syntax	Description
;DEBUG	This will display object rectangles in black and the touch areas in red. This allows the developer to see the positioning and sizing of each object.
;DEBUGOBJECTS	This will display object rectangles in black. Touch areas are not drawn
;DEBUGTOUCH	This will display object touch areas in red. Object rectangles are not drawn.
;DEBUGEXECUTION	This will output into the system log file the object execution path, including the line numbers of each object executed. Note that PocketWeather's debug logging must be enabled in PW Options/"General"/"Debug"
;RELOAD <icon image - optional> x,y	Causes the custom layout script to be reloaded

It is recommended that the script developer make use of the PocketWeather for PC/Win32 application to facilitate the writing and debugging of CustomViews.



Custom Layout Script - A Simple Example

Below is a simple example of a custom layout script which contains 1 objects and 1 additional header command which will be described later in this document.

```
// This is a CustomView script

;HEIGHT
45

;IMAGE
#ICON
10,10,20,20,c

;TEXT
#TEMP#
50,00,20,15,c

;TEXT
#HIGH#
50,10,20,15,c

;TEXT
#LOW#
50,20,20,15,c

;TEXT
Welcome to PocketWeather Scripting
50,30,190,15,c
```

This produces the following output





D-Pad Navigation & Gestures

Some OBJECTS (SUBVIEW, TOUCH) are associated with a TouchArea. The D-Pad "UP"/"DOWN" will highlight these areas sequentially allowing the D-Pad "ENTER" to execute the command that is associated with that TouchArea. "Right"/"Left" presses are also permitted and usually change to the next/previous Location. The currently selected Location is considered to be the "active" Location.

Other special buttons and commands (eg RADAR IMAGE) are also associated with Touch areas.

Gestures have recently been added where a "swiping" motion horizontally, vertically or diagonally will move the display as does "drop and drag" in some instances.



Objects

As described above, the multiline entity consist of the first line command

;COMMAND

which declares the object type, followed by a second line describing necessary attributes or elements (e.g. position) is termed an OBJECT.

The following table described the various objects that PocketWeather scripts commonly employ.

OBJECT Commands – Table 5

Command	Description
;TEXT	Displays text, variables, strings
;IMAGE	Displays images, maps, icons and other graphics
;DRAW	Displays shapes, lines and shaded areas
;SUBVIEW	Displays a defined panel
;TOUCH	Defines an action when a specified area of the display is pressed or selected
;GRAPH	Draws graphs
;IF	Conditional element
;TIME	Displays the time and/or date at the specified location using the specified font size and color

TEXT, IMAGE, DRAW, GRAPH, SUBVIEW, TOUCH, TIME and IF commands and their objects will be specifically defined subsequently in this manual.

If one wishes to prevent a particular command from executing, it can be disabled by converting its object into a comment.

All objects must be proceeded by an empty line AND carriage return and must be followed by an empty line, though one can place a commented title immediately above an object.

****Note** – when writing a script in NotePad, it is possible to have hidden formatting which will have unexpected results; e.g. you may think there is a carriage return before the start of an object, but there may not be one and the object will not execute properly.



Object Positioning and Dimensions

Most objects require positioning within a "window" defined by the starting point X,Y, which normally represents the upper left corner of the area of the object.

As well as X,Y positioning, objects may also require width and height dimensions, W, and H.

In addition to X,Y, W and H, some objects can have optional horizontal and vertical alignment specifier.

Syntax:

```
;TEXT
<AnyString>
<BoundingBox> , [<Alignment>]
[<FontSpecification>] , [<Color>] , [<Rotation>]
```

where BoundingBox is

X,Y, <Window Width>, <Window Height>

Example:

```
;TEXT
Some Text positioned 50,0, and width 200, and height 15, center aligned
50,0,200,15,cv
```

Dimensions can be absolute values, "W" or "H" for width or height, or percentage of dimension - e.g. 50% means 50% of width or height.

Example:

```
0, 0, 100%, 14
```

(starts at 0,0, 100% of width, 14 pixels height)

Example:

```
2%, 2%, 98%, 60%
```

(starts at 2% of width, 2% of height, and takes up remains of the width, and 60% of height)

Example:

```
10, 10, W-20,H-20
```

(starts at 10, 10, and takes up all the width and height, except for last 10 pixels horizontally and vertically).

Example:

```
2%+2,10%+100, 16, 16
```

(starts at 2% of width plus 2 pixels, 10% of height plus 100 pixel).



Alignment

Syntax:

```
;TEXT  
<AnyString>  
<BoundingBox>, [<Alignment>]  
[<FontSpecification>], [<Color>], [<Rotation>]
```

The following alignment attributes are allowed:

Horizontal/Vertical Alignment Attributes – Table 6

Attribute	Description
C	Center (horizontal) aligned
L	Left aligned
R	Right aligned
W	Word Break - cause text to go across multiple lines as appropriate
T	Top aligned
B	Bottom aligned
V	Vertical centered aligned
E	truncates any word that does not fit in the rectangle and adds ellipses ("...")

Note – Attributes can be combined where appropriate. Bottom, Top and VerticalCenter CANNOT be used with WordBreak, which only applies to text objects; the other attributes (except E) can apply to both text and graphics.

In addition, for ";TEXT" objects, the string may contain "\n", which is a "line feed" character to force a new line.

Example:

```
;TEXT  
This temperature display will be centered over two display lines \n #TEMP#  
50,20,80,30, WC
```



Font Specification

Specification of the font (the entire line of parameters) is optional; if all attributes are omitted, PocketWeather will use the default Today font, size and color. Otherwise, fonts can be specified using the following syntax with each attribute defined below. PocketWeather will use default attributes if any are omitted, but there can NOT be interposed omitted attributes (eg. if FontWeight is omitted, Color can not be specified and will be the default).

Syntax:

```
;TEXT  
<AnyString>  
<BoundingBox>, [<Alignment>]  
[<FontSpecification>], [<Color>], [<Rotation>]
```

Font Specification includes: [<FontName>], [<FontSize>], [<FontWeight>]

Font Attributes – Table 7

Attribute	Comment
Font Name	e.g. "Tahoma". Can be any name of fonts installed on the device. In addition, any TTF font file located in the same folder as the layout file can also be referenced.
Size	e.g. "12". The font point size
Font Weight	Valid attributes: <ul style="list-style-type: none">• "NORMAL"• "BOLD"• "UPPER" - uppercase characters• "LOWER" - lowercase characters• "SHADOW" - font filter• "GLOW-r-g-b" - font filter with r-g-b the color of the "glow" effect• "GLOWEXPAND-n" - n is the descriptor for the "glow" effect thickness• "ITALIC" - font style adornment• "UNDERLINE" - font style adornment

**** Note** – for PocketWeather, the term "FontWeight" (e.g. "**bold**") also includes FontStyle, e.g. "*italic*"; combinations are allowed (e.g. BOLD ITALIC UNDERLINE) as long as they are not mutually exclusive (e.g. BOLD NORMAL). Non-capitalized attributes may give unpredictable results.

Examples:

Tahoma,10,NORMAL,TODAY

TAHOMA, 10 BOLD ITALIC UNDERLINE,0-0-0,0

Bradley Hand ITC,13,UPPERCASE GLOW-100-51-100 GLOWEXPAND-2 BOLD,255-255-255



Colors

Color are specified using the standard R-G-B format.

Syntax:

[<R>-<G>-]

- R is the **red component** (0-255)
- G is the **green component** (0-255)
- B is the **blue component** (0-255)

where "0-0-0" produces black, and "255-255-255" produces white.

When producing custom views for the today screen, specifying "TODAY" will cause it to use the device's configured today screen text color.

NOTE: Colors MUST be specified as "R-G-B" - "R,G,B" is not accepted, and may cause unexpected results.

Examples:

12-34-56

Color Today/Home Screen Special Cases

The following special cases are supported:

- "TODAY" - uses the today/home screen default font color
- "HIGHLIGHT" - uses the today/home screen default highlight color
- "HIGHLIGHTTEXT" - uses the today/home screen default highlighted text color.

Default Colors

Omitting the color specifier will result in the following text font color being used:

- Today/Home Screen – uses the device's today/home screen text color as specified by the current color scheme
- WeatherConsole – always used black (0-0-0)

Rotation

Optional text rotation in degrees, where positive numbers increase the anti-clockwise angle. Default is 0. Negative numbers are not allowed.



Heading Commands

Header commands supply supplemental data at the beginning of the layout and may be valid for both TodayView and WeatherConsole or exclusive to one or the other. They are described in the table below.

Heading Commands – Table 8

Syntax	Description
;TITLE <some name>	For WeatherConsole, this specifies the name of the tab Ignored for Today/Home Screen views
;HEADING <Left-Text> <Right-Text>	WeatherConsole Only this specifies the text displayed on the top heading bar. Note that <Left-Text> and <Right-Text> can be any formatted text that can be used in the "Text Object". E.g. "#47001# #CITY# #TEMP#" Note that by default, it will use the current condition context, and if not available, the current forecast context. Ignored for Today/Home Screen views
;ICONID <IconID>	Should always be this for WeatherConsole. Ignored for Today Screen view
;ICON <Icon image>	Specify custom WeatherConsole tab icon. See "Icon Image" Note below.
;HEIGHT cy	The page height Must specify this for Today Screen. For WeatherConsole, if this is specified, then the user can scroll/pane vertically
;WIDTH cx	The page width No need to specify this for Today Screen. For WeatherConsole, if this is specified, then the user can scroll/pane horizontally. If you specify a width, then the view becomes fixed width and does not automatically resize
;SETTINGS <Updated icon image> <Updating icon image> <Out-Of-Date icon image> x,y	Draws the settings menu button at the specified position. Note that three icons need to be provided. See "Icon Image" Note below.



Syntax	Description
;DRAWTABS , <Size>, , <Color> <Tab Shading % - optional> <Tab Color - optional> <Icon Padding - optional> <Location Name Position> <"ANALOG"/"DIGITAL" Tab Clock - optional>, <Digital Tab Clock Position> <Digital Tab Clock Font Name>, <Size>, , <color>	See "DRAWTABS" section below
;NODRAWTABS	See "DRAWTABS" section below
;NODRAWTOPHEADINGBAR	WeatherConsole Only: disables the displaying of the top heading bar which displays the location name and date/time
;DEBUG	This will display object rectangles in black and the touch areas in red. This allows the developer to see the positioning and sizing of each object.
;DEBUGOBJECTS	This will display object rectangles in black. Touch areas are not drawn
;DEBUGTOUCH	This will display object touch areas in red. Object rectangles are not drawn.
;DEBUGEXECUTION	This will output into the system log file the object execution path, including the line numbers of each object executed.
;RELOAD <icon image> x,y	Causes the custom layout script to be reloaded. See "Icon Image" Note below.
;REFRESH <icon image> x,y	Causes weather and image update process to start. See "Icon Image" Note below.
;REFRESH-WEATHER <icon image> x,y	Causes weather update process to start. See "Icon Image" Note below.
;REFRESH-IMAGE <icon image> x,y	Causes image update process to start. See "Icon Image" Note below.
;UP <icon image> x,y	Location change – select previous location, display icon. See "Icon Image" Note below.
;DOWN <icon image> x,y	Location change – select next location, display icon. See "Icon Image" Note below.



Syntax	Description
;HIDPI	Flags that the layout script dimensions were designed for VGA/HiDPI devices. When running on a QVGA device, dimensions should be scaled appropriately. Note that PocketWeather automatically adjusts for QVGA/VGA displays and scales appropriately, and therefore this option can normally be ignored. Leaving this option out will result in the layout script being automatically scaled for both QVGA and VGA displays.
;ROTATE-START <icon image> x,y	Today Screen Only: Starts location rotation. See "Icon Image" Note below.
;ROTATE-STOP <icon image> x,y	Today Screen Only: Stops location rotation. See "Icon Image" Note below.
;ROTATE <icon image> x,y	Today Screen Only: Toggles location rotation. See "Icon Image" Note below.
;ROTATE-START-FORCE	Today Screen Only: Starts location rotation immediately on the initial run of the script. See "Icon Image" Note below.
;ROTATE-STOP-FORCE	Today Screen Only: Stops location rotation immediately on the initial run of the script. See "Icon Image" Note below.
;TEMP-UNITS-<units>	Specifies the default temperature units used within the script. This is independent of the configured PocketWeather temperature units. Valid units are: C – Celsius (default) F – Fahrenheit K – Kelvin
;PRESSURE-UNITS-<units>	Specifies the default pressure units used within the script. This is independent of the configured PocketWeather pressure units. Valid units are: mb – mBar (default) mm – mmHg in – inHg h – hPascal p – Pascal



Syntax	Description
<code>;SPEED-UNITS-<units></code>	Specifies the default speed units used within the script. This is independent of the configured PocketWeather speed units. Valid units are: mpg km (default) knot b – Beaufort ms – Meters/Second
<code>;DISTANCE-UNITS-<units></code>	Specifies the default distance units used within the script. This is independent of the configured PocketWeather distance units. Valid units are: k – Kilometers (default) f – Feet I – Inches n – Nautical Miles mm – Millimeters m – Meters mi – Miles
<code>;ELEVATION-UNITS-<units></code>	Specifies the default elevation units used within the script. This is independent of the configured PocketWeather elevation units. Valid units are: k – Kilometers (default) f – Feet I – Inches n – Nautical Miles mm – Millimeters m – Meters mi – Miles

****Note: "Icon Image"** - "<icon image>" is the name of the graphic file, either PNG, TGA, JPG, GIF, or BMP format.

Optionally, "<icon image>" can consist of two icons, the first is used for QVGA display devices, the second for VGA display devices. Icon Image.

Example:

```
;REFRESH
refresh.tga,refresh_vga.tga
10,10
```

Doing this will cause PocketWeather to select the appropriate icon depending on the device display configuration. On a QVGA device, "refresh.tga" will be used, and on a VGA device, "refresh_vga.tga".



DRAWTABS

Location Tabs - Today Screen Only (PPC/Classic/Professional devices only – i.e. Not SmartPhones/Standard)

The control of the today screen location tabs is described in summary in the previous Heading section. This section provides additional details.

Without any layout command, Location Tabs will appear with default values controlled only by the configured options in PocketWeather Options under "Today".



(Default – no clock or weather icons)



(Default – analog clock and weather icons)

To explicitly disable tabs from appear, use the **”;NODRAWTABS”** command in the header.

Syntax:

”;NODRAWTABS

Note that if the **”;NODRAWTABS”** command is specified, then the SETTINGS button MUST also be specified. See **”;SETTINGS”** in Table 8 above.

To explicitly enable tabs to appear, use the **”;DRAWTABS”** command in the header.

Syntax:

”;DRAWTABS

, <Size>, , <Color>

<Tab Shading %>

<Tab Color >

<Icon Padding>

<Location Name Position>

<"ANALOG"/"DIGITAL" Tab Clock>, <Digital Tab Clock Position>

<Digital Tab Clock Font Name>, <Size>, , <Color>

**DrawTab Commands – Table 9**

Syntax	Description
<FontName>, <Size>, <FontWeight>, <Color>	see above section on fonts
<Tab Shading %>	As percent. If tab shading is >0, then unselected location tabs are shared. If tab shading is <0, then selected location tab is shared
<Tab Color>	Tab color – if not specified, then uses default color.
<Y Padding> OR <X, Y Padding>	The tab Padding is affects the total tab width and height – and is the additional width and height added to the minimum tab width and height required by the settings button icon, font height and width, and analogue/digital clock. Note that if only one value is specified (I.e. "20"), then for legacy reasons, this is taken to be the vertical padding. * See Padding Note below
<LocationNamePosition>	Specifies the position of the location name within the tab. See above section on "Object Positioning and Dimensions". Valid position specifiers, "T,V,B,L,C,R"
<TabClock>, <DigitalTabClockPosition>	"ANALOG" - displays analogue clock to the left of the location's name. The Analogue clock will ALWAYS be displayed on the far left. "DIGITAL" - displays a digital clock. Default position is below and to the right of the location's name. See above section on "Object Positioning and Dimensions". Valid position specifiers, "T,V,B,L,C,R" "NOCLOCK" - disables the display of clock.
<DigitalTabClockFontName>, <DigitalTabClockFontSize>, <DigitalTabClockFontweight>, <DigitalTabClockFontColor>	see above section on font
<TabWeatherIcon>, <TabWeatherIconAlignment>	"FORECASTICON" - displays the current day's weather icon "CURRENTICON" - displays the current condition's weather icon Can also specify position. See above section on "Object Positioning and Dimensions". Valid position specifiers, "T,V,B,L,C,R" Note that if the height of the weather icons is greater than the height of the tab, the weather icon will not be displayed. The currently configured "Mini" icons will always be used. If the weather icon fails to appear, increase the Y padding.

Padding Note – PocketWeather is unable to guess the desired layout being designed, and therefore will only ensure minimum horizontal space to display the location's name and the optional clock (digital or analogue). Therefore the script writer MUST ensure enough vertical and horizontal space using the X-Y padding to allow space for the location name, option clock, and weather icon. If undesirable overlapping occurs, increase the X and/or Y padding.



Example:

```
;DRAWTABS  
Tahoma,10,BOLD,TODAY  
28  
TODAY  
16,0  
RT  
DIGITAL,RB  
Tahoma,10,N,TODAY  
CURRENTICON,LV
```



Example:

```
;DRAWTABS  
Courier New,12,BOLD ITALIC  
UNDERLINE,TODAY  
28  
TODAY  
0  
RV  
ANALOGUE
```





Image Objects

Syntax:

```
;IMAGE
<ImageTypes>
X, Y, W, H, <Horizontal alignment>
[ROTATION-<Variable>] [<LimitHigh> <LimitLow>] [<RangeHigh> <RangeLow>]
```

ImageTypes – Table 10

ImageTypes	Comment
<Filename>	must be BMP, TGA, JPG, GIF or PNG file
<Filename-QVGA>,<Filename-VGA>	will auto select the QVGA/VGA BMP/TGA/PNG/GIF/JPG file depending on device's current DPI setting
#ICON or #DAY-ICON	the WeatherIcon for the configured context (FORECAST/CURRENT); appending "-WIND" causes WindArrow (WindVector) to be overlayed. Adding "NOWIND" disables/prevents WindArrow from being drawn
#RADAR-ALL	displays all the downloaded images enabled for that location. Note that "RADAR" images are automatically allocated Touch Objects which link to WeatherConsole. See section on Touch Objects below.
#RADAR-n	displays the n'th downloaded image for that location - zero based. Note that "RADAR" images are automatically allocated Touch Objects which link to WeatherConsole. See section on Touch Objects below.
#CURRENTICON	the CurrentCondition WeatherIcon; appending "-WIND" causes WindArrow (WindVector) to be overlayed. Adding "NOWIND" disables/prevents WindArrow from being drawn
#MOON	the current day's moon phase icon
#WIND	displays the wind compass and WindArrow (WindVector)
#WINDARROW	displays just the wind direction arrow
#nnn#	displays Image from the PocketWeather Standard Image Resources where "nnn" is the ImageID. See " Standard Image IDS – Table 11 " below.



Icon Sizes

With the following command:

```
;IMAGE  
#ICON  
0,0
```

it will automatically select the default configured icon.

If you want to explicitly control the icon size, this can be achieved using the following specifiers after the "#ICON" keyword to select the different configured icon sizes

- MINI - 16x16 for QVGA, 32x32 for VGA
- NORMAL - 32x32 for QVGA, 64x64 for VGA
- LARGE - 64x64 for QVGA, 128x128 for VGA
- GIANT - 96x96 for QVGA, 192x192 for VGA

Example:

```
;IMAGE  
#ICON-MINI  
0,0
```

```
;IMAGE  
#ICON-LARGE  
30,0
```



Day/Night Icon

With the following command:

```
;IMAGE  
#ICON  
0,0
```

it will automatically select the correct day/night icons depending on the location's local time.

By default, all forecast weather icons use day-time icons.

You can forecast the use of day or night icons by added "-DAY" or "-NIGHT" to the icon command.

Example:

```
;IMAGE  
#ICON-DAY  
0,0
```

```
;IMAGE  
#ICON-MINI-NIGHT  
0,0
```

```
;IMAGE  
#ICON-LARGE-DAY  
30,0
```



Optional Rotation Specification

Syntax:

```
;IMAGE
<Image object type>
X, Y, W, H, <Horizontal alignment>
[ROTATION-<Variable>] [<LimitHigh> <LimitLow>] [<RangeHigh> <RangeLow>]
```

Where <Variable> is scaled between <LimitLow> and <LimitHigh> to map between <RangeLow> and <RangeHigh>.

<Variable>, <LimitLow>, <LimitHigh>, <RangeLow> and <RangeHigh> can be any numerical value or any variable listed in the "Supported Variables" Table 13.

<RangeLow> represent the starting rotation point in degrees, and <RangeHigh> represents the end rotation point in degrees, where vertically up is 0 degrees; working clockwise, 0 through to 360 degrees completes the rotation.

To configure a rotation which starts at 270 degrees (9th hour, pointing left) and ends at 90 degrees (3rd hour, pointing right), passing through 0 degrees (12th hour, vertically up), the RangeHigh/RangeLow configuration must be "450 270". The increasing value of RangeHigh implies a clockwise direction. If RangeHigh/RangeLow configuration were "90 270", because of the decreasing value of RangeHigh, a counter-clockwise (anti-clockwise) direction would be implied.

The defaults are:

```
<HighLimit> 60
<LowLimit> 0
<RangeHigh> 360
<RangeLow> 0
```

Example:

Rotation-TIME-MINUTES

cause the image to be rotate by the number of minutes of the current clock.

Example:

Rotation-TIME-HOURS 0 12

































The following code produces the example shown

```
;IMAGE
hand-qvga.png,hand-vga.png
0,0,W, H, CV
ROTATE-BAROMETER 1050 950 497 228
```





























Note that for images to rotate correctly, they MUST be square. That means the images height MUST be the same as its width.




















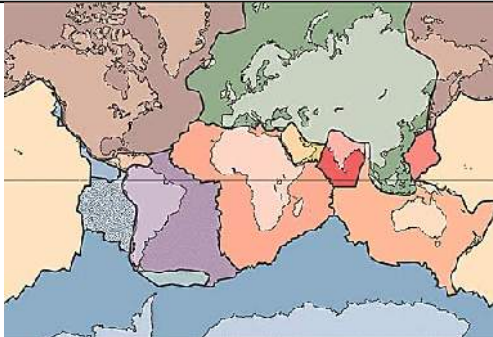
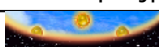



**Standard Image IDS – Table 11**

ID	Description	QVGA Images	VGA Images
101	Pocket Weather Logo non-transparent		
102	Pressure Icon		
104	Visibility Icon		
105	Dewpoint Icon		
106	Precipitation Icon		
107	Thermometer Icon		
108	Map Zoom In		
109	Map Zoom Out		
110	Map Rotate		
111	Map animation controller – play		
112	Map animation controller – pause		
113	Left arrow		
114	Right arrow		
115	Selected Icon		
116	WeatherConsole Title Icon		
117	Wind Arrow 1 (small)	 (wind_arrow_1.png)	 (wind_arrow_vga_1.png)



ID	Description	QVGA Images	VGA Images
118	Wind Arrow 2 (medium)	 (wind_arrow_2.png)	 (wind_arrow_vga_2.png)
119	Wind Arrow 3 (large)	 (wind_arrow_3.png)	 (wind_arrow_vga_3.png)
120	Compass	 (compass.png)	 (compass_vga.png)
121	WeatherConsole Earthquake Tab Icon	 (earthquakes_tab.png)	 (earthquakes_tab_vga.png)
122	WeatherConsole Summary Tab Icon	 (Summary_tab.png)	 (Summary_tab_vga.png)
123	WeatherConsole METAR Tab Icon	 (METAR_tab.png)	 (METAR_tab_vga.png)
124	WeatherConsole Forecast Tab Icon	 (Forecast_tab.png)	 (Forecast_tab_vga.png)
125	WeatherConsole Forecast Compare Tab Icon	 (Forecasts_tab.png)	 (Forecasts_tab_vga.png)
126	WeatherConsole Maps Tab Icon	 (maps_tab.png)	 (maps_tab_vga.png)
127	WeatherConsole Maps Compare Tab Icon	 (mapscom_tab.png)	 (mapscom_tab_vga.png)
130	Folder Selection Dialog – Up Icon	 (up.png)	 (up_vga.png)
131	Folder Selection Dialog – Toolbar Icon	 (toolbar.png)	 (toolbar_vga.png)
132	WeatherConsole Home Icon	 (home.png)	 (home_vga.png)
133	Current Temperature Icon	 (temp.png)	 (temp_vga.png)



ID	Description	QVGA Images	VGA Images
140	Down Arrow Icon		
141	Up Arrow Icon		
150	Settings Button Up-To-Date	 (updated.png)	 (Updated_vga.png)
152	Settings Button Out-Of-Date	 (Outdated.png)	 (Outdated_vga.png)
153	Settings Button Updating	 (Updating.png)	 (Updating_vga.png)
160	Alarm Bell		
161	Alert Icon		
200	Earthquake Map	 (worldmap.jpg)	 (worldmap_vga.jpg)
201	Earthquake Tectonic Map	 (worldmap2.jpg)	 (worldmap2_vga.jpg)
300	Sun Rise/Set Background image		
400	PocketWeather Config main image list		

** NOTE - depending on the device's DPI settings, using these will automatically select the VGA or QVGA image size



Draw Objects

Syntax:

```
;DRAW
```

```
RECTANGLE PENCOLOR-<PENCOLOR> BRUSHCOLOR-<BRUSHCOLOR> ALPHA-<Shading>  
WIDTH-<Line Width>
```

```
X, Y, W, H
```

or

```
;DRAW
```

```
LINE PENCOLOR-<PENCOLOR> WIDTH-<Line Width>
```

```
X, Y, X-Direction, Y-Direction
```

or

```
;DRAW
```

```
CIRCLE PENCOLOR-<PENCOLOR> BRUSHCOLOR-<BRUSHCOLOR> ALPHA-<Shading>  
WIDTH-<Line Width>
```

```
X, Y, W, H
```

Example:

```
;DRAW
```

```
RECTANGLE PENCOLOR-0-0-0 BRUSHCOLOR-250-250-240 WIDTH-2  
0%,14,100%,64
```

or

```
;DRAW
```

```
LINE PENCOLOR-0-0-0 WIDTH-2  
40%, 14, 0, 64
```

or

```
;DRAW
```

```
RECTANGLE PENCOLOR-0-0-0 BRUSHCOLOR-250-250-240 WIDTH-2 ALPHA-50  
0%,14,100%,64
```

This draws a rectangle 50% shaded/blended with the background.

Note that for line objects, the X/Y directions can be positive or negative.

Note that for "CIRCLE" draw objects, if the height is the same as the width, then a circle is drawn. Otherwise an ellipse is produced.

For "RECTANGLE" draw objects, if the height is the same as the width, then a square is drawn.



Alpha-Shading

Syntax:

;DRAW

*RECTANGLE PENCOLOR-<PENCOLOR> BRUSHCOLOR-<BRUSHCOLOR> ALPHA-<Shading>
WIDTH-<Line Width>*

X, Y, W, H

where <Shading> is a percentage from 0% through to 100%.

Shading of 0% results in NO change to the background, while a shading of 100% results in all the background being replaced by "BRUSHCOLOR". Values between 0% and 100% result in variable blending of the background with "BRUSHCOLOR".

Note that the process of blending the background with "BRUSHCOLOR" is CPU processor intensive and will result in extended view rendering time, slow response of the device, and reduction of battery life. This is particularly true for larger objects and on VGA devices.

Therefore it is recommended that this feature be used wisely.



Draw Objects With Variable Size

The simple draw objects described in the previous section can have their dimensions modified at runtime based on weather variables such as current temperature, wind speed, pressure, etc.

The syntax for these draw objects with variable size is identical to the normal draw objects, with the additional of extra attributes. These attributes can apply to both lines and rectangles.

In addition to the rectangle and line color and width attributes, the following options attributes can be used to change the height, width or length of the objects at run time based on weather variables such as current temperature, wind speed, etc.

Attributes – Table 12

Name	Description
VARIABLE-<constant/variable>	Declares the value or constant to use. For example, current temperature. See "Supported Variables – Table 13" below
VARIABLEMAX-<constant/variable>	Declares the maximum range of the variable. This can be a fixed constant value, or another value such as temperature high limit. See "Supported Variables – Table 13" below
VARIABLEMIN-<constant/variable>	Declares the minimum range of the variable. This can be a fixed constant value, or another value such as temperature low limit. See "Supported Variables – Table 13" below
VARIABLEDIRECTION-<direction>	Defines which direction to scale. Valid entries are "UP", "DOWN", "LEFT" and "RIGHT. For example, "VARIABLEDIRECTION-UPDOWNLEFTRIGHT", or " VARIABLEDIRECTION-UP", or " VARIABLEDIRECTION-RIGHT". The default is "UP". For line draw objects, the default operation is to always start at the defined starting point, and to scale to length from that point in the specified direction.

Example:

```
;DRAW
LINE PENCOLOR-121-163-187 WIDTH-10 VARIABLE-TEMP VARIABLEMIN-LOW
VARIABLEMAX-HIGH
0,14,100%,0
```

This would result in a line starting on the left-hand side and extending horizontally to the right, where the length would be based on the current temperature, between the limits of the high and low temperatures. So if the current temperature was 20°C, the minimum temperature was 10°C, and the maximum temperature was 30°C, the line would start on the left, and extend 50% (I.e. Half of 100%) across to the right.

Example:

```
;DRAW
RECTANGLE PENCOLOR-121-163-187 WIDTH-10 VARIABLE-TEMP VARIABLEMIN-LOW
VARIABLEMAX-HIGH VARIABLEDIRECTION-RIGHT
0,0,100%,100%
```

This would result in a rectangle starting in the top left corner, extending from the top to the bottom fully, but only extending to the right by the amount defined by the current temperature within the range of the forecast high and low temperatures.



Draw Line With Variable Angle

Syntax:

```
;DRAW
LINE PENCOLOR-<PENCOLOR> WIDTH-<Line Width> LENGTH-<Line Length constant>
VARIABLEMIN-<constant/variable> VARIABLEMAX-<constant/variable> ROTATIONMAX-
<constant/variable> ROTATIONMIN-<constant/variable> VARIABLEDIRECTION-
<constant/variable>
X, Y
```

The *<VARIABLEDIRECTION>* attribute allows a line to be rotated about a given starting point.

The rotation angle depends on *<VARIABLEDIRECTION>*, which is scaled between *<VARIABLEMIN>* and *<VARIABLEMAX>*, which maps between *<ROTATIONMIN>* and *<ROTATIONMAX>*.

<VARIABLEDIRECTION>, *<VARIABLEMIN>*, *<VARIABLEMAX>*, *<ROTATIONMIN>* and *<ROTATIONMAX>* can be any numerical value or any variable listed in the "Supported Variables" Table 13.

<ROTATIONMAX> represent the starting rotation point in degrees, and *<ROTATIONMIN>* represents the end rotation point in degrees, where vertically up is 0 degrees; working clockwise, 0 through to 360 degrees completes the rotation.

To configure a rotation which starts at 270 degrees (9th hour, pointing left) and ends at 90 degrees (3rd hour, pointing right), passing through 0 degrees (12th hour, vertically up), the *<ROTATIONMAX>* / *<ROTATIONMIN>* configuration must be "450 270". The increasing value of *<ROTATIONMAX>* implies a clockwise direction. If *<ROTATIONMAX>* / *<ROTATIONMIN>* configuration were "90 270", because of the decreasing value of *<ROTATIONMAX>*, a counter-clockwise (anti-clockwise) direction would be implied.

The defaults are:

```
<VARIABLEMAX> 60
<VARIABLEMIN> 0
<ROTATIONMAX> 360
<ROTATIONMIN> 0
```

Example:

```
;DRAW
LINE PENCOLOR-200-200-200 WIDTH-2 LENGTH-92 VARIABLEMIN-0 VARIABLEMAX-60
VARIABLEDIRECTION-TIME-MINUTE
50%,50%

;DRAW
LINE PENCOLOR-200-200-200 WIDTH-2 LENGTH-75 VARIABLEMIN-0 VARIABLEMAX-12
VARIABLEDIRECTION-TIME-HOUR
50%,50%
```

**Supported Variables – Table 13**

Name	Description	Associated Weather Template Variable
CITYID	The location's index ID	
BATTERYTEMP	the device's current battery temperature - availability depends on device's hardware	
BATTERYLIFTPERCENTAGE	the device's current battery life percentage - availability depends on device's hardware	
TEMP	temperature (current, or forecast, as appropriate)	"tmpr" and "tmprUnits"
FEELSLIKETEMP	effective temperature	(calculated)
HUMIDITY	Relative Humidity	"humidity"
DEWPOINT	Dew point	"dewPoint" and "tmprUnits"
RAIN	Rain Probability % (I.e. change of rain falling in the future)	"precipitation"
RAINFALLAMOUNT	Amount of rain recorded (I.e. what has already fallen) in Meters.	"rainfall" and "rainfallUnits"
RAINFALLDAYS	Number of days with precipitation	"rainfallDays"
SUNSHINEHOURS	Monthly average number of hours of sunshine per day	"sunshineHours"
BAROMETER	the barometer value	"pressure" and "pressureUnits"
VISIBILITY	Visibility distance	"visibility" and "visibilityUnits"
WINDDIR-NUMBER	wind direction in degrees - e.g. 0' to 360'	"windDirection"
WINDSPEED	Wind speed	"windSpeed" and "windSpeedUnits"
WINDGUSTSPEED	Wind gust speed	"windGustSpeed" and "windSpeedUnits"
HIGHAPPARENT	apparent forecast high/max/day temperature	(calculated)
LOWAPPARENT	apparent forecast low/min/night temperature	(calculated)
HIGH	forecast high/max/day temperature	"tmprHi" and "tmprUnits"
LOW	forecast low/min/night temperature	"tmprLo" and



Name	Description	Associated Weather Template Variable
		"tmprUnits"
MOON-PERCENT	moon phase %	(calculated)
UV	UV (sun stength) Index	UvIndex
DPI	The device's display DPI	
DISPLAYWIDTH	<p>The panel view's width.</p> <p>Note that this is real/actual display width, AFTER the effect of display DPI scaling for VGA and SmartPhone devices where the device's DPI setting is different from 96DPI.</p> <p>When used to compare against a constant, the following macro should be used:</p> <p>"DISPLAYWIDTH(<value>)"</p> <p>Example:</p> <p>;IF-DISPLAYWIDTH >= DISPLAYWIDTH(240)-BEGIN</p>	
DISPLAYHEIGHT	<p>The panel view's height.</p> <p>Note that this is real/actual display height, AFTER the effect of display DPI scaling for VGA and SmartPhone devices where the device's DPI setting is different from 96DPI.</p> <p>When used to compare against a constant, the following macro should be used:</p> <p>"DISPLAYHEIGHT(<value>)"</p> <p>Example:</p> <p>;IF-DISPLAYHEIGHT >= DISPLAYHEIGHT(320)-BEGIN</p>	
TIME-YEAR	The current year (e.g. 2009)	
TIME-YEAR-LOCAL	The current year in location local time (e.g. 2009)	
TIME-MONTH	The current month (1 = January, 2 = February...)	
TIME-MONTH-LOCAL	The current month in location local time	
TIME-DAY	The current day	
TIME-DAY-LOCAL	The current day in location local time	
TIME-DAYOFWEEK	The current day of week (Sunday = 0, Monday = 1, etc)	
TIME-DAYOFWEEK-LOCAL	The current day of week in location local time (Sunday = 0, Monday = 1, etc)	



Name	Description	Associated Weather Template Variable
TIME-HOUR	The current hour. Note – fraction includes portion of hour (i.e. minute/60)	
TIME-HOUR-LOCAL	The current hour in location local time. Note – fraction includes portion of hour (i.e. minute/60)	
TIME-MINUTE	The current minute. Note – fraction includes portion of minute (i.e. second/60)	
TIME-MINUTE-LOCAL	The current minute in location local time. Note – fraction includes portion of minute (i.e. second/60)	

Note that “TEMP” (current temperature) is a current conditions value, and normally, “HIGH”, “LOW”, “HIGHAPPARENT” and “LOWAPPARENT” are forecast values. So unfortunately you cannot use “HIGH” and “LOW” to set the range (max/min) of “TEMP”. This restriction may be removed in the future.

Note that for variables with associated units, the following default units are assumed:

Temperature	Degrees C
Pressure	mBar
Speed	KPH
Visibility Distance	Kilometres
Elevation	Meters
Depth	Kilometres

Therefore, when specifying a variable with fixed limits, you should assume the about units and use the appropriate units.

To change the default units, see “;TEMP-UNITS-<units>” heading above.

Temperature Bar Example:

```
;DRAW
RECTANGLE PENCOLOR-121-163-187 WIDTH-10 VARIABLE-TEMP VARIABLEMIN--10
VARIABLEMAX-40 VARIABLEDIRECTION-RIGHT
0,0,100%,100%
```



Historical Variables

The following historical variables are valid.

Historical Variables – Table 14

Name	Description	Associated Weather Template Variable
HISTORICALRAINFALLAMOUNT	Amount of rain recorded (I.e. what has already fallen) in Meters.	"rainfall" and "rainfallUnits"
HISTORICALRAINFALLDAYS	Number of days with precipitation	"rainfallDays"
HISTORICALSUNSHINEHOURS	Monthly average number of hours of sunshine per day	"sunshineHours"
HISTORICALHIGH	high/max/day temperature	"tmprHi" and "tmprUnits"
HISTORICALLOW	low/min/night temperature	"tmprLo" and "tmprUnits"

Note that the availability of historical values depends on the appropriate historical template feed providing the necessary data.



Graph Object

Syntax:

```
;GRAPH  
<COMMANDS>  
X, Y, W, H
```

The "<COMMANDS>" can contain any of the following commands:

Graph Commands – Table 15

Type Command (only one can be specified)	Description
TEMP	Displays temperature graph
RAIN	Displays rain trending (subject to availability from weather feed)
HUMIDITY	Displays humidity trending (subject to availability from weather feed)
BAROMETER	Displays barometric pressure trending (subject to availability from weather feed)
ICON	Displays weather icons for each day and any significant weather change.
LEGEND	Just draws the graph background legend.

Only one of the above commands can be specified.

Each of these commands can be have the following display attributes – note that conflicting attributes are ignored (I.e. DAILY and HOURLY...)

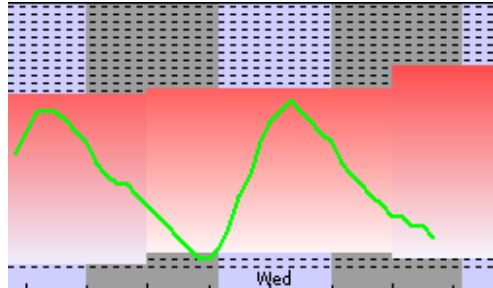
Display Attributes – Table 16

Display Attributes	Description
DAILY	Disables the hourly forecast
HOURLY	Enables the hourly forecast
"N"HOURLY	Displays the next N hours – e.g. "24hours". Note that there MUST be no gap between the number of hours and the word "HOURS"
"N"DAYS	Displays the next N days - .e.g. "5days". Note that there MUST be no gap between the number of days and the word "DAYS"
NOLEGEND	Disables graph background legend display
COLOR-R-G-B	R-G-B color of the graph line.



Example:

```
;GRAPH  
TEMP 48HOURS COLOR-0-255-0  
0,0,100%,100%
```



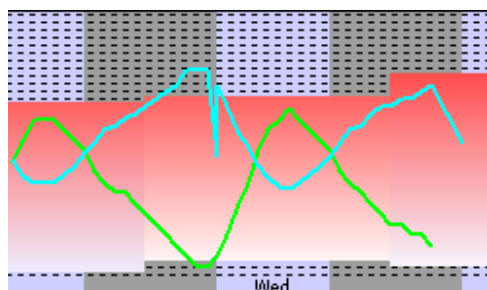
Using the "LEGEND" command, you can build up map overlays.

Example:

```
;GRAPH  
48HOURS LEGEND  
0,0,100%,100%
```

```
;GRAPH  
TEMP 48HOURS COLOR-0-255-0 NOLEGEND  
0,0,100%,100%
```

```
;GRAPH  
HUMID 48HOURS COLOR-0-255-255 NOLEGEND  
0,0,100%,100%
```





Text Object

Syntax:

```
;TEXT  
<The String>  
X,Y,W,H,<Alignment>  
<Font Name>,<Size>,<Font weight>,<Color>
```

<The String> can be any free format string.

In addition, "\n" is the "line feed" character used to force a new line.

```
;TEXT  
This temperature display will be centered over two display lines\n#TEMP#  
50,20,80,30,WC
```

The following tables list the variable reserved.

Date/Time Objects

Note:

1. ALL times and dates are assumed to be in device/user local time unless explicitly specified.
2. ALL components refer to the currently active location unless specified or not appropriate.
3. ALL time and date components refer to the currently active location, and to the currently selected weather item (i.e. current conditions, or forecast data).
4. "#DATE" variables, when applied to daily forecasts, will not have any UTC to user/device/location local time conversion applied, because they always refer to the forecast date. In such cases, "#DATE-DAYOFWEEK-LOCAL#" and "#DATE-DAYOFWEEK#" will always display the same string regardless of the timezone the user/device is in and the location's timezone.
5. "#TIME" variables – appending "-NODATE" will cause the time string to never include the date. e.g. "#TIME-NODATE"
6. "#TIME" variables – appending "-HOURSONLY" will cause the time string to ONLY include the hours. e.g. "#TIME-HOURSONLY"

**Time and Date Variables – Table 17**

Name	Description
#UPDATE-TIME#	the most recent update time, whatever was most recently updated (i.e. current conditions, forecast, images)
#UPDATE-TIME-FORECAST#	the last forecast update time
#UPDATE-TIME-CURRENT#	the last current conditions update time
#UPDATE-TIME-IMAGES#	the last images update time
#UPDATE-TIME-EARTHQUAKE#	The last earthquake update time
#UPDATE-SCHEDULE-NEXT#	The next scheduled update – scheduler must be enabled
#UPDATE-TIME-NEXT#	The next update time. If the scheduler is enabled, then displays the next earliest scheduled update time, otherwise the next update time based on the existing updates and the appropriate update rates.
#UPDATE-TIME-HISTORICAL#	The last historical update time
#TIME#	The current time
#TIME-LOCAL#	The current time in location local time
#TIME-METAR#	The time when the METAR observation data was recorded
#TIME-METAR-LOCAL#	The time when the METAR observation data was recorded in location local time.
#TIME-FORECAST#	The current forecast time. Only meaningful for hourly forecast data.
#TIME-FORECAST-LOCAL#	The current forecast time in location local time. Only meaningful for hourly forecast data. (see note 4 above)
#TIMEZONENAME#	The location's current timezone name, as defined by the operating system
#TIMEZONEID#	The locations's current timezone ID. This is the operating system's unique identifier.
#TIMEZONE#	The location's current timezone offset in fractions of hours.
#TIMEZONEABB#	The location's current timezone name as an abbreviation. This is generated from the full name.
#DATE-DAYOFWEEK#	Day of week number – e.g. "1" (Monday = 1, etc)
#DATE-DAYOFWEEK-LOCAL#	Day of week number in location local time – e.g. "1" (Monday = 1, etc) (see note 4 above)
#DATE-DAYOFWEEK-STRING#	Brief version of the day string – e.g. "Mon"
#DATE-DAYOFWEEK-STRING-LOCAL#	Brief version of the day string in location local time – e.g. "Mon" (see note 4 above)
#DATE-DAYOFWEEK-STRING-LONG#	Long version of the day string – e.g. "Monday"
#DATE-DAYOFWEEK-STRING-LONG-LOCAL#	Long version of the day string in location local time – e.g. "Monday" (see note 4 above)
#DATE-DAY#	Day of month
#DATE-DAY-LOCAL#	Day of month in location local time (see note 4 above)
#DATE-DAY-LONG#	Day of month with leading "0" if appropriate
#DATE-DAY-LONG-LOCAL#	Day of month in location local time with leading "0" if appropriate (see note 4 above)
#DATE-MONTH#	Month number



Name	Description
#DATE-MONTH-LOCAL#	Month number in location local time (see note 4 above)
#DATE-MONTH-LONG#	Month number with leading "0" if appropriate
#DATE-MONTH-LONG-LOCAL#	Month number in location local time with leading "0" if appropriate (see note 4 above)
#DATE-MONTH-STRING#	Short Month string – e.g. "Jan"
#DATE-MONTH-STRING-LOCAL#	Short Month string in location local time – e.g. "Jan" (see note 4 above)
#DATE-MONTH-STRING-LONG#	Month string – e.g. "January"
#DATE-MONTH-STRING-LONG-LOCAL#	Month string in location local time – e.g. "January" (see note 4 above)
#DATE-YEAR-LONG#	Year number– e.g. "2010"
#DATE-YEAR-LONG-LOCAL#	Year number in location local time– e.g. "2010" (see note 4 above)
#DATE-YEAR#	Year number– e.g. "08"
#DATE-YEAR-LOCAL#	Year number in location local time– e.g. "08" (see note 4 above)
#DATE-LONG#	The long version of the date string as specified by the device's regional settings and any user customization
#DATE-LONG-LOCAL#	The long version of the date string in location local time as specified by the device's regional settings and any user customization (see note 4 above)
#DATE#	The short version of the date string as specified by the device's regional settings and any user customization
#DATE-LOCAL#	The short version of the date string in location local time as specified by the device's regional settings and any user customization (see note 4 above)



General Variables

Note that the following variables are either result from the specific location's configuration, or derived/calculated.

General Variables – Table 18

Name	Description
#METARSTATIONNAME#	the currently selected METAR station name, or ICAO code if the name is not available
#METARSTATIONICAO#	the currently selected METAR station ICAO code
#METAR#	Outputs the full raw METAR data for the currently selected ICAO station
#METAR-n#	Outputs the full raw METAR data for ICAO station "n"
#TAF#	Outputs the full raw TAF data (short term forecast) for the currently selected ICAO station
#TAF-n#	Outputs the full raw TAF data (short term forecast) for ICAO station "n"
#CITY#	Displays the location name
#CITYID#	Displays the location's ID – this is the location's position within the user's list of locations.
#CITYCODE#	Displays the location's country code
#CITYCOUNTRY#	Displays the location's country name
#ASTRONOMICALTWILIGHTSTART#	Twilight Start (before sunrise) - sun is 18' below the horizon
#NAUTICALTWILIGHTSTART#	Twilight Start (before sunrise) - sun is 12' below the horizon
#CIVILTWILIGHTSTART#	Twilight Start (before sunrise) - sun is 6' below the horizon
#TWILIGHTSTART#	Civil Twilight Start (before sunrise) - same as CIVILTWILIGHTSTART
#SUNRISE#	sun rise
#SUNNOON#	Noon – when the sun is at its highest
#SUNSET#	sun set
#TWILIGHTEND#	Twilight End (after sunset) - same as CIVILTWILIGHTEND
#CIVILTWILIGHTEND#	Twilight End (after sunset) - sun is 6' below the horizon
#NAUTICALTWILIGHTEND#	Twilight End (after sunset) - sun is 12' below the horizon
#ASTRONOMICALTWILIGHTEND#	Twilight End (after sunset) - sun is 18' below the horizon
#SUNHOURS#	number of daylight hours
#MOON-PERCENT#	moon phase %
#MOON-TEXT#	moon phase text
#S-DEG#	the temperature units string
#S-SPEED#	the speed units string
#S-DIST#	the distance units string
#S-RAINFALLAMOUNT#	the depth units string
#nnnnn#	extracts the appropriate language resource string from PocketWeather's language resource strings - definition of IDs is below
#BATTERYTEMP#	the device's current battery temperature - availability depends on device's hardware



Name	Description
#BATTERYLIFTPERCENTAGE#	the device's current battery life percentage - availability depends on device's hardware

Current Conditions Weather Template Related Variables

Note that despite the following variable being available, not all current conditions weather feeds provide all entries. Please check with the particular current conditions weather feeds as to which of the following provide meaningful data.

Current Condition Variables – Table 19

Name	Description	Associated Weather Template Variable
#TEXT#	the current weather description text (i.e. current conditions, or appropriate day's forecast)	Derived from "skyTextID", or if not available, "sky"
#TEMP#	temperature (current, or forecast, as appropriate)	"tmpr"
#FEELSLIKETEMP#	Effective temperature	(calculated)
#HUMIDITY#	Relative Humidity	"humidity"
#DEWPOINT#	Dew point	"dewPoint"
#RAINFALLAMOUNT#	The amount of rain fall recorded – includes units	"rainfall" and "rainfallUnits"
#BAROMETER-TEXT#	"Rising"/"Steady"/"Falling"	(calculated)
#BAROMETER-SYMBOL#	A symbol representing the rising/falling/steady state	(calculated)
#BAROMETER#	the barometer value	"pressure" and "pressureUnits"
#VISIBILITY#	Visibility distance	"visibility" and "visibilityUnits"
#UV#	UV (sun strength) Index	"uvIndex"
#WINDDIR-TEXT#	wind direction text	"windDirection"
#WINDDIR-NUMBER#	wind direction in degrees - e.g. 0' to 360'	"windDirection"
#WINDSPEED#	Wind speed	"windSpeed"
#WINDSPEEDUNITS#	Wind speed, plus appends the appropriate units. If no wind, displays "Calm"	"windSpeedUnits"
#WINDGUSTSPEED#	Wind gust speed	"windGustSpeed"
#WINDGUSTSPEEDUNITS#	Wind gust speed, plus appends the appropriate units.	"windSpeedUnits"



Forecast Weather Template Related Variables

Note that despite the following variable being available, not all forecast weather feeds provide all entries. Please check with the particular forecast weather feeds as to which of the following provide meaningful data.

Forecast Variables – Table 20

Name	Description	Associated Weather Template Variable
#TEXT#	the current weather description text (i.e. current conditions, or appropriate day's forecast)	Derived from "skyTextID", or if not available, "sky"
#HUMIDITY#	Relative Humidity	"humidity"
#DEWPOINT#	Dew point	"dewPoint"
#RAIN#	Rain Probability	"precipitation"
#WINDDIR-TEXT#	wind direction text	"windDirection"
#WINDDIR-NUMBER#	wind direction in degrees - e.g. 0' to 360'	"windDirection"
#WINDSPEED#	Wind speed	"windSpeed"
#WINDSPEEDUNITS#	Wind speed, plus appends the appropriate units. If no wind, displays "Calm"	"windSpeedUnits"
#WINDGUSTSPEED#	Wind gust speed	"windGustSpeed"
#WINDGUSTSPEEDUNITS#	Wind gust speed, plus appends the appropriate units.	"windSpeedUnits"
#HIGH#	forecast high/max/day temperature	"tmprHi"
#LOW#	forecast low/min/night temperature	"tmprLo"
#HIGHLOW#	forecast max/min formatted appropriate	"tmprHi" and "tmprLo"
#HIGHAPPARENT#	apparent forecast high/max/day temperature	(calculated)
#LOWAPPARENT#	apparent forecast low/min/night temperature	(calculated)
#FORECASTUSERTEXT#	Displays the contents of the "forecastUserText" variable provided by the weather feed template. Note that this is specific to each forecast data entry – different representations of the variable will exist for each hourly and daily forecast entries.	"forecastUserText"
#UV#	UV (sun strength) Index	"uvIndex"



Historical Weather Template Related Variables

Note that despite the following variable being available, not all locations support historical data, and not all entries. Please check with the particular historical weather data as to which of the following provide meaningful data.

The following variables are only available when used within a ";HISTORICAL" loop – see "Selecting Historical Data" section below.

Historical Variables – Table 21

Name	Description	Associated Weather Template Variable
#RAINFALLDAYS#	Number of days with precipitation	"rainfallDays"
#SUNSHINEHOURS#	Monthly average number of hours of sunshine per day	"sunshineHours"
#RAINFALLAMOUNT#	The amount of rain fall recorded – includes units	"rainfall" and "rainfallUnits"
#HIGH#	forecast high/max/day temperature	"tmprHi"
#LOW#	forecast low/min/night temperature	"tmprLo"

To access the historical data for the currently selected current conditions or forecast section (I.e. within a ";CURRENT" or ";FORECAST" loop – see "Selecting Weather Data" section below), use the following variables.

Historical Variables Within Weather Loops – Table 22

Name	Description	Associated Weather Template Variable
#HISTORICALRAINFALLDAYS#	Number of days with precipitation	"rainfallDays"
#HISTORICALSUNSHINEHOURS#	Monthly average number of hours of sunshine per day	"sunshineHours"
#HISTORICALRAINFALLAMOUNT#	The amount of rain fall recorded – includes units	"rainfall" and "rainfallUnits"
#HISTORICALHIGH#	forecast high/max/day temperature	"tmprHi"
#HISTORICALLOW#	forecast low/min/night temperature	"tmprLo"

Prepending "HISTORICAL" before the variable will cause PocketWeather to automatically try and find the matching historical data for the currently selected date.

Note that the availability of historical values depends on the appropriate historical template feed providing the necessary data.



Weather Alert Template Related Variables

Note that despite the following variable being available, not all weather feeds provide all entries. Please check with the particular weather feeds as to which of the following provide meaningful data.

Weather Alert Variables – Table 23

Name	Description	Associated Weather Template Variable
#ALERT-FORECAST#	Displays forecast alert message – depends on feed and configuration. This will be generated by the METAR feed	Extracted from METAR feed
#ALERT-CURRENT-CONDITIONS#	Displays any and all severe current condition weather alerts for all the location's METAR stations - depends on alert configuration.	Extracted from METAR feed
#ALERT-CURRENT-CONDITIONS-n#	Displays any severe current condition weather for a particular METAR station, where n is the zero based station index.	Extracted from METAR feed
#ALERT-n#	Displays the content of the "alertn" variable provided by the weather feed template, where "n" is "0", "1"...	"alertTextn"
#ALERT#	Displays the content of all alert strings available (forecast, current conditions, or "alertn" variables, depending on the weather feed.	"alertText"
#USER-n#	Displays the content of the "user-n" variable provided by the weather feed template, where "n" is "0", "1"...	"userTextn"

Earthquake Variables

The following variables are extracted from the fixed earthquake feed.

Earthquake Variables – Table 24

Name	Description
#EARTHQUAKE-DATE#	displays date of latest earthquake
#EARTHQUAKE-MAG#	displays magnitude of latest earthquake
#EARTHQUAKE-LOC#	displays location of latest earthquake
#EARTHQUAKE-NEAR#	displays nearest town/city to latest earthquake
#EARTHQUAKE-n-DATE#	displays date of latest-n earthquake
#EARTHQUAKE-n-MAG#	displays magnitude of latest-n earthquake
#EARTHQUAKE-n-LOC#	displays location of latest-n earthquake
#EARTHQUAKE-n-NEAR#	displays nearest town/city to latest-n earthquake



Debugging Variables

The following variables can be used to help debug a script.

Debugging Variables – Table 25

Name	Description
#LINENUMBER#	Displays the script line number
#OBJECTINDEX#	Displays the object's internal index
#OBJECTDEPTH#	Displays the object's nesting depth level
#VIEWNAME#	Displays the full path and file name of the layout script

Example:

```
;TEXT  
#TEMP#°#S-DEG#  
0,14,20%,10,C
```

```
;TEXT  
#30010#  
80%,15,20%,12,c
```

String IDS

These are automatically translated into the configured language.

Example:

```
;TEXT  
#30015#  
0,14,20%,10,C
```

For more details of available string IDs, see the "Strings" section.



Location Command Object

These objects are used to modify the current working set within the custom layout.

Location Command – Table 26

Name	Description
;LOCATION-NEXT	Selects the next enabled location
;LOCATION-PREVIOUS	Selects the previous enabled location
;LOCATION-HOME	Selects the location flagged as the home location
;LOCATION-n	Selects the nth enabled location
;LOCATION-<name/ID>	Selects the location with specified name, or specified country ID (e.g. "UKXX0080", "London")
;LOCATION	Remain with the same location

Note that when a location command is encountered, all previous states are reset – i.e. Position reset to far left, selected forecast back to today.

Example:

;IMAGE

#Icon#

0, 0

;LOCATION-NEXT

;IMAGE

#Icon#

0, 20

The above will display the weather icon for the selected location, then the weather icon for the next enabled location.



Time Object

Syntax:

<Time Command>

X,Y,W,H,<Alignment>

[],[<FontSize>],[<FontWeight>],[<Color>],[<Rotation>]

This object is used for displaying the time and/or date at the specified location using the specified font size and color.

****Note -**

1. ALL times and dates are assumed to be in device/user local time unless explicitly specified.
2. Adding "-LOCAL" will result in the time and dates being in the currently selected location's local time, e.g. ";TIME-LOCAL"
3. Unlike the Date and Time formats for the ";TEXT" objects, this object always used the current time
4. Unless specified, all times and dates are displayed using the device's configured date and time format. This is controlled by the "Regional Settings" in the device's settings "System" panel.

Time Command Attributes – Table 27

Available Time Commands	Description
;TIME	Displays the current time (only)
;TIME-TIMEANDDATE	Displays the current date and time
;TIME-DAYOFWEEK-STRING-LONG	Displays the day of week as a long string – e.g. "Thursday"
;TIME-DAYOFWEEK-STRING	Displays the day of week as a short string – e.g. "Thu"
;TIME-DAYOFWEEK	Displays the day of week as a number – e.g. "1" (Monday = 1, etc)
;TIME-DAY-LONG	Displays the day of month as a 2 digit number – e.g. "01"
;TIME-DAY	Displays the day of month as either a 1 or 2 digit number – e.g. "1", or "28"
;TIME-MONTH-STRING-LONG	Displays the month as a long string – e.g. "November"
;TIME-MONTH-STRING	Displays the month as a short string – e.g. "Nov"
;TIME-MONTH-LONG	Displays the month as a 2 digit number – e.g. "01"
;TIME-MONTH	Displays the month as either a 1 or 2 digit number – e.g. "1", or "11"
;TIME-YEAR-LONG	Displays the year as a 4 digit number – e.g. "2008"
;TIME-YEAR	Displays the year as a 2 digit number – e.g. "08"
;TIME-DATE-LONG	Displays the full date as a long string – e.g. "21 November 2008". Note that the format is controlled by the device's "Regional Settings".
;TIME-DATE	Displays the full date as a string – e.g. "21/11/2008". Note that the format is controlled by the device's "Regional Settings".

Example:

;TIME-DAYOFWEEK-STRING-LOCAL

0,7,60,15,cte

Tahoma,8,GLOW-255-255-255 GLOWEXPAND-1,0-0-0



Sub Views Object

These objects are used to include the contents of another file

Syntax:

```
;SUBVIEW  
<custom layout file>  
X,Y,W,H  
<optional touch command when object selected and activated – (I.e. tapped)>
```

Example:

```
;SUBVIEW  
somecustomlayoutfile.pcx  
0,40,100%,40
```

The above will display the contents of the file "somecustomlayoutfile.pcx" at the location 0, 40, width 100%, height 40

Note that it takes the location context from any previous commands. So:

```
;SUBVIEW  
multiline-single.pwx  
0,35,W,30  
weatherconsole.exe #CITYID#  
  
;LOCATION-NEXT  
  
;SUBVIEW  
multiline-single.pwx  
0,60,W,30  
weatherconsole.exe #CITYID#
```

the above will process the contents of multiline-single.pwx, then switch to the next location, and then display the contents of multiline-single.pwx again, which will be for the second location.

If the user taps on the region defined for the subview, then the application "weatherconsole.exe" will be executed with the cityid passed as a command line argument. The application is first searched within the PocketWeather main installation folder, then the "\\Windows" folder, then the device's system paths (I.e. Lets the operating system try and find it.).

Refer to section on Touch objects for fuller details of the touch commands.



Touch Object

These objects are used to include the contents of another file

Syntax:

```
;TOUCH  
X,Y,W,H  
<optional touch command when object selected and activated – (I.e. tapped)>
```

Example:

```
;TOUCH  
0,35,W,30  
weatherconsole.exe #CITYID#
```

If the user taps on the defined touch region, then the application "weatherconsole.exe" will be executed with the cityid passed as a command line argument. The application is first searched within the PocketWeather main installation folder, then the "\\Windows" folder, then the device's system paths (I.e. Lets the operating system try and find it.).

See later section of details of weatherconsole command line options.

The same text expansions as defined for the Text Objects are available.

Touch commands can also be used to cause the view to switch to another custom script layout file.

Example:

```
;TOUCH  
0,35,W,30  
page1.pwc
```

Note that the custom script layout file MUST be local to the current file's folder.



Touch Area Ordering

Touch areas overlay previously declared touch areas in the following order:

Last In, First Out

This means that if a touch area covering the entire view is declared first, then a subsequent touch area also covering the entire view is declared second, the second one will hide the first one.

Example:

```
// touch area 1 – links to page 1 - covering entire view (yellow)
```

```
;TOUCH
```

```
0,0,W,H
```

```
page1.pwc
```

```
// touch area 2 – links to page 2 – covering left side of view (red)
```

```
;TOUCH
```

```
0,0,50%,H
```

```
page2.pwc
```

```
// touch area 3 – links to page 3 – covers middle of view (green)
```

```
;TOUCH
```

```
25%,0,50%,H
```

```
page3.pwc
```

```
// touch area 4 – links to page 4 – covers right side of view (blue)
```

```
;TOUCH
```

```
50%,0,50%,H
```

```
page4.pwc
```

The result would be as follows:



Touch area 1 (yellow) is completely hidden by areas 2, 3, and 4.



WeatherConsole Command Line Options

Valid WeatherConsole command line options are:

weatherconsole.exe <CITYID> <FULL PATH TO LAYOUT SCRIPT>

or

weatherconsole.exe <CITYID> <TABID> <SUBTABID>

where:

WeatherConsole Command Line Options – Table 28

TABID Function	SUBTABID	EXAMPLE
0 - Custom Layout	0 – First Custom View (default) 1 – 2 nd Custom View 2 – 3 rd Custom View ... (depends on number of custom views currently configured in WeatherConsole)	<i>Weatherconsole.exe #CITYID# 0 0</i> <i>Weatherconsole.exe #CITYID# 0 3</i>
1 - METAR Map	0 – Map View 1 – Station Data View 2 – Station Compare	<i>Weatherconsole.exe #CITYID# 1</i> <i>Weatherconsole.exe #CITYID# 1 2</i>
2 - Location Forecast	0 – Summary 1 – Details 2 – Sun and Moon 3 - Hourly	<i>Weatherconsole.exe #CITYID# 2 1</i>
3 - Location Images	Image ID	<i>Weatherconsole.exe #CITYID# 3</i> <i>Weatherconsole.exe #CITYID# 3 4</i>
4 - Forecast Compare	0 – Forecast Summary 1 – Current Summary 2 – Current Details 3 – Temperature 4 – Precipitation 5 – Humidity 6 – Icons 7 – Wind Speed & Direction 8 – Daylight Times 9 – Trendings	<i>Weatherconsole.exe #CITYID# 4</i> <i>Weatherconsole.exe #CITYID# 4 3</i>
5 - Images Compare		<i>Weatherconsole.exe #CITYID# 5</i>
6 - Earthquakes/World View	0 – World Map 1 - Details	<i>Weatherconsole.exe #CITYID# 6</i> <i>Weatherconsole.exe #CITYID# 6 1</i>

When specifying the full path to a layout script, the first custom layout tab will always be selected. Note that the setting of the new custom layout script for the first custom layout tab is only temporary – navigating away and back will revert the first custom layout tab back to its configured script.



Conditional Statements

Syntax:

```
;IF[-NOT]-<CONDITION>-BEGIN
```

```
.
```

```
.
```

```
.
```

```
[/ELSE]
```

```
.
```

```
.
```

```
.
```

```
[/END]
```

or

```
;IF-<CONDITION>
```

```
;<OBJECT>
```



Conditional Statements - Conditions

The following conditions can be used

Conditional Statements – Table 29

Name	Description
FORECAST	If there is forecast weather data available
FORECAST-HOURLY	If there is hourly forecast weather data available
CURRENT	If there is valid and up-to-date current condition weather data available
METAR	If the current conditions data is from a METAR station
HISTORICAL	If there is historical data available
RADAR-n	If the selected location has a configured image icon "n" available
DISPLAY-APPARENT	If the apparent temperature is different from the actual temperature for the currently selected weather
DAY-DD	DD = Day of Week 0 = Sunday 1 = Monday 2 = Tuesday 3 = Wednesday 4 = Thursday 5 = Friday 6 = Saturday
DATE-YYYY-MM-DD	YYYY = year MM = Month DD = Day of Month
TIME-HH-MM-SS	HH = Hours (24-hour clock) MM = Minutes (optional) SS = Seconds (optional) ";IF-TIME-22" would trigger when the hour is 10pm
WINDGUST	If wind gusts are recorded
ALERT-FORECAST	If there is a weather forecast alert for the currently selected location
ALERT-CURRENT-CONDITIONS	If there is a current conditions alert for the currently selected location at any of the location's configured METAR stations
ALERT-CURRENT-CONDITIONS-n	If there is a current conditions alert for METAR station "n" for the currently selected location.
ALERT-n	If the "alertn" weather template feed variable has provided a string, where "n" is "0", "1"...
ALERT	If any of the "alert" variables provide a string
USER-n	If the "usern" weather template feed variable has provided a string, where "n" is "0", "1"...
USER	If any of the "user" variables provide a string
ANY-ALERT	If there is either a weather forecast, current conditions, or weather template feed variable.



Name	Description
DAYTIME	True if between sunrise and sunset
NIGHTTIME	True if between sunset and sunrise
TWILIGHT	Same as CIVILTWILIGHT
CIVILTWILIGHT	True if between civil twilight start and sunrise, or sunrise and civil twilight end
NAUTICALTWILIGHT	True if between nautical twilight start and sunrise, or sunrise and nautical twilight end
ASTRONOMICALTWILIGHT	True if between astronomical twilight start and sunrise, or sunrise and astronomical twilight end
BP-RISING	If barometric pressure is rising
BP-FALLING	If barometric pressure is falling
BP-STEADY	If barometric pressure is steady
PRESSURE (or BP)	If barometric pressure is available
<LeftValue> <OPERATOR> <RightValue>	See Conditional Operators section below.
VGA	If device's display DPI is VGA (> 140 DPI)
QVGA	If device's display DPI is QVGA (< 140 DPI)
QVGASP	If device's display DPI is QVGA AND running on SmartPhone device (DPI is 132)
LANDSCAPE	If Panel's width is greater than its height

Example:

```
;IF-CURRENT-BEGIN
```

```
;TEXT
```

```
#TEMP#
```

```
0,45,40,15,c
```

```
;ELSE
```

```
;TEXT
```

```
No Current condition data available
```

```
0,45,40,15,c
```

```
;ENDIF
```

or

```
;IF-RADAR-0
```

```
;IMAGE
```

```
#RADAR-0#
```

```
40,0, 40, 57, cv
```



Conditional Operators

Syntax:

<LeftValue> <OPERATOR> <RightValue>

****WARNING**** you MUST ensure a space BETWEEN the <LEFT-VALUE> and the <OPERATOR>, and BETWEEN the <OPERATOR> and <RIGHT-VALUE>.

**** Note** – make sure the correct units have been declared in the Header (c.f. Header Commands above)

The <LeftValue> and <RightValue> can be any numerical value or any variable listed in the "Supported Variables" Table 13.

Example:

;IF-TEMP > 20-BEGIN

Conditional Operators – Table 30

Operator	Description
">"	Greater than. Note that PocketWeather requires there to be a differences of a least "1.0" for this condition to be true. Example: ;IF-1.5 > 2.4 This will return FALSE (because 2.4 – 1.5 = 0.9) Example: ;IF-1.5 > 2.6 This will return TRUE (because 2.6 – 1.5 = 1.1)
">="	Greater or Equal
"<="	Less or Equal
"<"	Less than. Note that PocketWeather requires there to be a differences of a least "1.0" for this condition to be true. Example: ;IF-1.5 < 2.4 This will return FALSE (because 2.4 – 1.5 = 0.9) Example: ;IF-1.5 < 2.6 This will return TRUE (because 2.6 – 1.5 = 1.1)
"="	Equal to
"!="	Not Equal
"<>"	



Conditional Statements – “NOT”

Any of the above conditions can be logically inverted (inversed, negative) using the “NOT” operator. The example below shows how to determine when the current conditions data is NOT valid:

```
;IF-NOT-CURRENT-BEGIN  
  
;TEXT  
No Current condition data available  
0,45,40,15,c  
  
;ELSE  
  
;TEXT  
Current conditions is valid – temperature is #TEMP#  
0,45,40,15,c  
  
;ENDIF
```

The following code produces the same results

Conditional NOT Examples – Table 31

Example 1	Example 2
<pre><i>;IF-TIME-11-BEGIN</i> <i>;ELSE</i> <i>;TEXT</i> <i>The Time is NOT 11am</i> <i>0,45,40,15,c</i> <i>;ENDIF</i></pre>	<pre><i>;IF-NOT-TIME-11-BEGIN</i> <i>;TEXT</i> <i>The Time is NOT 11am</i> <i>0,45,40,15,c</i> <i>;ENDIF</i></pre>



CUSTOM Positioning

Syntax:

```
;CUSTOM[-ABSOLUTE]  
X, Y
```

This command allows you to set/reset the starting position to the specified X, Y position when displaying any object following the next scripted ";FORECAST" and ";HISTORICAL" section. This is needed when displaying forecast data using the ";FORECAST" object.

Note that the custom positioning is relative to previous positions, unless you specify "-ABSOLUTE", when the positions become absolute.

Example:

```
;CUSTOM-ABSOLUTE  
40,15
```



Selecting Weather Data Context

Forecast Data

Syntax:

```
;FORECAST
<X-Offset>,<Y-Offset>,<IterationCount>,<IncrementCount>,<StartFrom>

.
.

;FORECAST-END
```

This sets the weather data for Forecast context, and will cause the subsequent objects/commands to be executed for each forecast day starting from the current forecast day.

On each iteration, the display position will be offset by the specified <X-Offset> and <Y-Offset> values.

A total of <IterationCount> loops will be executed, or until the display position is off the displayable view.

If no <IterationCount> is specified, then it will loop until the display position is off the displayable view.

If <IncrementCount> is specified, then on each iteration, the forecast skips the specified number of entries. For hourly forecast, <IncrementCount> refers to the number of hours to skip, while daily forecasts, <IncrementCount> refers to the number of days to skip. An <IncrementCount> of zero (0) is the same as one (1).

If <StartFrom> is specified, then the forecast iteration will explicitly offset the forecast day/hour by the specified offset (day/hour). If no <StartFrom> is specified, then the forecast day/hour will continue on from any previous iteration, or if none, will start from the beginning.

When parsing the subsequent objects/commands following the ";FORECAST" command, all objects/commands until the end of the file, or the first ";FORECAST-END" statement (which ever comes first) will be executed.

Specifying the ";FORECAST-END" statement allows for multiple loops or lists of forecast days to be handled. If a subsequent ";FORECAST" command is declared, the selected forecast day continues from where the previous loop ended, unless that new ";FORECAST" command explicitly declares a <StartFrom> value.

For compatibility, the following special case is supported

```
;FORECAST
<Y-Offset>
```

This will assume a zero <Y-offset>, and no iteration count – all forecast days will appear horizontally until the display position is off the displayable view.

Note that for objects with display positioning, they automatically inherit the X-offset accumulated by the previous ";FORECAST" and ";CUSTOM" object declaration.



Example	Result
<pre>;FORECAST 40 ;TEXT #DAY# 0,0,40,15,c ;IMAGE #ICON 0,9,40,40,cv ;TEXT #HIGHLOW# 0,45,40,15,c ;TEXT- #RAIN#% 0,54,40,15,c ;FORECAST-END</pre>	
<pre>;FORECAST 30,40,4 ;TEXT #DAY# 0,0,40,15,c ;IMAGE #ICON 0,9,40,40,cv ;TEXT #HIGHLOW# 0,45,40,15,c ;FORECAST-END ;CUSTOM-ABSOLUTE 80,0 ;FORECAST 30,40,4 ;TEXT #DAY# 0,0,40,15,c ;IMAGE #ICON 0,9,40,40,cv ;TEXT #HIGHLOW# 0,45,40,15,c ;FORECAST-END</pre>	



Forecast For Specific Day

Syntax:

```
;FORECAST-n
```

This sets the weather data context to forecast for the "n"th day, where 0 is the current day, 1 is tomorrow, and so on.

Since only one day is specified, no ";FORECAST-END" statement is needed.

Example:

```
;FORECAST-0
```

```
;IMAGE
```

```
#ICON
```

```
0,9,40,40,cv
```

```
;FORECAST-1
```

```
;IMAGE
```

```
#ICON
```

```
40,9,40,40,cv
```

```
;FORECAST-2
```

```
;IMAGE
```

```
#ICON
```

```
0,49,40,40,cv
```

```
;FORECAST-3
```

```
;IMAGE
```

```
#ICON
```

```
40,49,40,40,cv
```

Note that:

```
;FORECAST-n
```

is the same as:

```
;FORECAST
```

```
<X-Offset>,<Y-Offset>,<IterationCount>,<IncrementCount>,n
```



Forecast From Today

Syntax:

;CURRENT-FORECAST

This will select the current day's forecast context.

Example:

;CURRENT-FORECAST

;TEXT

#DAY#

0,0,40,15,c

Note that this is identical to specifying ";FORECAST-0".

Forecast From Tomorrow

Syntax:

;SKIP-FIRST-FORECAST

When used in with ";FORECAST", this causes it to skip the current day's forecast and iterate through the remaining days.

Example:

;SKIP-FIRST-FORECAST

;FORECAST

40

;TEXT

#DAY#

0,0,40,15,c

Note that this is identical to specifying ";FORECAST-1".



Hourly Forecast

Syntax:

```
;FORECAST-HOURLY  
  
.  
.  
  
;FORECAST-END
```

This causes it to iterate through the available hourly forecast, starting from the currently selected forecast, and repeating the objects/commands that follow, until it runs out of space, where the width of each hourly entry is the numeric value below – in this case “40”.

The optional “;FORECAST-END” flags the end of the looping for each hour. This allows for the script to handle multiple loops or lists of forecast data. If a subsequent “;FORECAST-HOURLY” command is declared, the selected forecast hour continues from where the previous loop ended.

Example:

```
;FORECAST-HOURLY  
40  
  
;IMAGE  
#ICON  
0,9,40,40,cv  
  
;FORECAST-END
```

To display a specific hourly forecast, use

```
;FORECAST-HOURLY-<n>  
40
```

where n represents the number of hours to skip forecast. So “;FORECAST-HOURLY-24” skips forward 24 hours.



Selecting Historical Data

PocketWeather V2.1.1 and has the ability to support historical data. This is data obtained from various websites which provide an annual overview of monthly temperatures, precipitation, etc.

This historical data can be accessed using the same syntax as the ";FORECAST" commands, but instead, using:

```
;HISTORICAL  
<X-Offset>,<Y-Offset>,<IterationCount>
```

```
.  
.
```

```
;HISTORICAL-END
```

In this form, PocketWeather will iterate from January through to December.

See the above section on ";FORECAST" for details of the syntax.

To access the current month's historical data, use the following syntax:

```
;HISTORICAL-0
```

For the next month, use

```
HISTORICAL-1;
```



Current Conditions

Syntax:

;CURRENT

This selects the current conditions context.

;CURRENT

;IMAGE

#ICON

0,9,40,40,cv