



The CMX Alternative Interface (AI)

The CumulusMX AI is an alternative to the supplied Interface for the system. It is intended to be used in addition to rather than as a replacement for the supplied interface. This is to ensure that you always have the access provided and developed by Mark and updated by him as required. Upgrades to the AI will be provided as soon as feasible when upgrades are developed by Mark but no guarantee is given or implied that the AI will provide all the features that Mark develops.

This version is intended for CumulusMX version 3.26.0 build 3256. It will NOT work on earlier builds.

I have still not been able to make the 'Settings ~ Internet Settings ~ Web/Upload Site ~ General Settings OR Advanced Settings' tabs fully expand. If these need adjusting you will have to use the default interface.

Installation

The supplied zip file contains this document and a separate zip containing the AI files. Please read this document for information relating to the changes and functionality of the this version.

Please remember that this is **not** a replacement for the supplied interface. If you use it as a replacement, it will be lost when you next upgrade CumulusMX.

- Unzip the ai2 packed to its own folder not part of your CumulusMX installation.
- Copy the ai2 folder into the existing interface folder for CumulusMX making sure it's a subfolder of it; I.e. CumulusMX/interface/ai2.

Warning if you don't make it a sub-folder, you will corrupt the default interface; if you don't copy it to the interface folder, it simply will not work.

- You should now be able to access the AI using:
 - i. localhost:8998/ai2/
 - ii. <ip address>:8998/ai2/
- You may need to clear your cache to prevent old versions of the support files being used by mistake.



The folder structure is:

- ai.cmx - The html files for this upgrade. I have changed the name of some of these files. This includes a 'development.html' page that you can use for experimentation and development!!
- /css Just the Main style sheet files
 - /img This contains modified images
 - /js Modified versions of all the default interface JavaScript files. Generally this involves ensuring that API calls use the root URL. Some additional changes have been made to pages using alpaca forms to omit the font stylesheets and where I have used buttons rather than the structures used by Mark. I have changed the name of some of these files to match their parent html file.
 - / lib contains the following full or partial libraries
 - datatables:** Fixed columns and a modified alt editor script.
 - HighCharts:** The 'ow-theme.js' file for charts
 - steelseries:** Contains the full library but only the 'gauges.js' file is required as this has had the tooltips modified to use the themes.
 - /themes contains the theme style sheets.

Any other files that the AI uses are accessed from the main interface folder so are unchanged.

Changes

Below I detail the changes I have made. Generally, this is confined to styles used to format the pages, to small changes made to the JavaScript files that generate each page and to a very small number of the library files.

Sitewide

The site now uses Font Awesome icons which have been added to the library. No internet connection is needed. The free version is supplied but includes the pro icons if you have a license.

The fonts are now fully responsive and adjust correctly without any jumps in size. See Appendix A for details.



All 'panels' on all pages use flex boxes whether they are a single column or multiple. This means that the layout of all pages can be amended to suite your system. All flex box panels have an order style that controls the order in which the panels are displayed. It also manages the row and column gaps between panels.

Wherever possible tables have been removed but due to the way that data is made available this is not always possible.

The seagull image that sits at the bottom right of the screen is animated but can have alternative animations applied using the AI-Configure page.

A new set of themes are provided - incompatible with previous versions but held separately. It is intended to release a new alternative public website that will also use these themes shortly.

A number of 'animation' styles are available and can be applied to any element. The timing for them can be edited in the stylesheet or, for the seagull, on the configuration page. To change the timing of the animations you will need to edit the 'main.css' stylesheet which is clearly documented to make this as easy as possible.

The menus are now animated to drop down and the 'Settings' menu is three or two column so that it doesn't disappear off the bottom of small screens.

The Dashboard

This page includes both a 'Davis' information panel and the alarms. The Davis panel does not appear if you don't have a Davis weather station connected and both panels toggle so once hidden, they stay hidden for the current browsing session.

If you do use a Davis station, you can add further battery status LEDs for additional remote sensors. This is documented in the html code.

The graphics have been undated and extended to include rain rate, (four alternative icons depending on the rate).

Current Readings

The 'Current Readings' page is slightly more detailed showing additional wind values. It also includes the trend graphics found on the Dashboard. FontAwesome icons are used in the title bars of each panel.

Again, all panels are part of a flex-box and can be displayed in any order just by using the order style. The default column layout is controlled by the flex-box declaration and can be changed to 'row-oneCol' if preferred - all data will continue to display correctly.

Gauges

The gauges are now the same size and style as on the default Interface. Each panel is a member of a flex-box so can be rearranged simply by changing their order number.

Charts

Firstly on all charts, the buttons, the chart and the 'chart description text' are part of a flex-box structure so, as with all pages so far, you can place each of these elements in the order you prefer. This also applies to the buttons.

All charts are zoomable, you can select sub-sets of data along the 'x' axis.

The standard charts use a button set rather than the dropdown used on the default Interface. These buttons expand to fill rows depending on the data you can graph. The active chart button text is always different to the text of other buttons. They also remember the chart you were last viewing in the current browser session.

The charts all use a locally stored theme javascript file so that they use the site theme colours. The charts plot colours are as configured in CumulusMX. If you don't want the charts to be configured in this way then simply don't link the 'ai-theme.js' file at the bottom of the HighCharts section of the page head.

One advantage of using the theme, however, is that you can configure your own colours for the 'Select-a-Chart' and 'Select-a-Period' charts. See Appendix 1 for details.

Finally, the charts use a height based on the height of your screen (with a minimum of 450px). This means the charts will resize when your screen height changes.

Today Vs Yesterday

This is a problem page as it is constructed using tables. My default design uses a two column flex box but if preferred, this can be changed to a one column style to avoid text and data spilling onto multiple lines. See Appendix 1 for details.

Weather Records

This is a single column layout using a flex box structure so it can be set at two columns if required. However, title and data will spill onto multiple lines. Being a flex box structure you can re-order the panels as required.

NOAA Reports

Both NOAA Reports automatically load the selected report when either the year or month selectors change. This means that the 'Load' button is not really needed. It can be removed by editing the page code and adding 'w3-hide' to its class list.

Extra & AirLink Sensors

Both pages use a two column flex box to organise the page but this can be changed to a one column display if preferred. All panels display tables of data. Many panels have FontAwesome icons in their title.

The 'Extra Sensors' page uses some additional table styles to highlight the second column but this can easily be removed or expanded as required. See Appendix 1 for details.

Data Logs, Extra Data Logs & The DayFile Editors

All these pages use an additional stylesheet to configure the popup data editor window. This has been extensively changed to give it a maximum height (based on your screen height and to subsequently scroll to display all data. The gaps between the data rows has been changes as has the column distribution widths.

The buttons within the dataTables framework are styled in the dedicated stylesheet - 'ai-editabletables.css' so if you have issues with specific theme colours for the buttons or the text, that is the place to look. The original issue with buttons not being fully visible in dark mode theme has been fixed.



As for previous AI's there are styles for units used within the table. I have not attempted to link this to your CumulusMX units so you will need to edit the html document for the units that you use. This should only affect temperature, rainfall, windspeed and pressure.

Extra Web Files

I have arbitrarily set a minimum width for the table used on this page but it does scroll sideways if required.

MySQL Settings

By default set as a two column page but works as one column but does not fill the site width for some reason. Edit line 120 to 'ow-oneCol' if this is required.

Monthly Records Editor

As ever, I use buttons to select the months rather than the tabs that Mark uses. These follow the format used throughout the site and also toggle in that it remembers which month you were looking at while you remain on the site. The current month is highlighted.

All Settings pages

All these pages now work as aspected and in the same way as on the default interface.

Edit record pages

These have been remodelled.

Utilities

All the utilities are listed in the menu structure. However, all utilities are available on just one page - utilities.html. (The 'Latest Errors' log is still available separately).

If you use the default database options and MySQL errors occur you will find that this page spills over on the right due to there being no spaces in the embedded mySQL statements. I am looking into programatically adding spaces after every comma to ensure that the commands wrap correctly.



AI Configure

This is now a three column page but includes all the needed configuration options. If you use the default website, you can also display (and copy) a suitable configuration object for this as well. (Currently in development).

Appendix A

Adjusting fonts

Responsive fonts appears to be a dark arts subject with many proposals. My solution is shown below:

```
body, p, label {  
    font-family: Inter, serif;  
    font-size: calc( 14px + 1 * (( 100vw - 300px) / ( var( --siteWidth ) - 300 )));  
}  
h1 { font-size: calc( 24px + 6 * (( 100vw - 300px) / ( var(--siteWidth) - 300 ))); }  
h2 { font-size: calc( 20px + 5 * (( 100vw - 300px) / ( var(--siteWidth) - 300 ))); }  
h3 { font-size: calc( 18px + 4 * (( 100vw - 300px) / ( var(--siteWidth) - 300 ))); }  
h4 { font-size: calc( 16px + 3 * (( 100vw - 300px) / ( var(--siteWidth) - 300 ))); }  
h5 { font-size: calc( 14px + 2 * (( 100vw - 300px) / ( var(--siteWidth) - 300 ))); }  
h6 { font-size: calc( 13px + 1 * (( 100vw - 300px) / ( var(--siteWidth) - 300 ))); }
```

The main values to edit are the first two - the base pixel size and the enlargement factor. Taking h1 as an example, the 24px represents the smallest size the font can be while 30px (24px + 6), is the largest it can be. The remaining calculation adjusts the enlargement factor based on the screen width. I believe the 300px represents the smaller screen width allowed and so the font will only be 24px high when the screen is 300px wide.

Page columns

All pages use flex box children to display columns. These classes: 'ow-colOne' to 'ow-colSix' control the minimum and maximum width of the child 'div' elements. These are calculated based on the site width. However, as screen width reduces, some are modified to be 100% width to ensure the maximum screen width is used. Using the style 'flex: 1;' ensures that they expand to the maximum available.

As they are children of flex boxes, they can all have an 'order' style. This can be used to override the position of the child within the box. Children with the same order number are displayed in the sequence they are coded in the page, otherwise the order is as dictated by the order number.

Charts

All the charts now use the theme selected for the site. This is supplied in the 'HighCharts' library area. This file can be edited but you should remember that it can only be used to change styles.

One feature of this script is that it defines the eight colours used by HighCharts on the 'select-a-chart-' pages. Therefore you can freely edit these colours as required. Colours used on the other charts are determined by CumulusMX. I have not investigated using the CumulusMX colours here but it should be possible.

Editing Themes

All themes are easily edited. They start with a 'root' section that defines the colours used. This is shown below:

```
:root {
  --sub5: #f5f5f5;
  --sub4: #dddddd;
  --sub3: #bbbbbb;
  --sub2: #989898;
  --sub1: #767676;
  --theme: #555555;
  --add1: #4c4c4c;
  --add2: #434343;
  --add3: #3b3b3b;
  --add4: #323232;
  --add5: #2a2a2a;
  --modal: rgba(85, 85, 85, 0.3);
}
```

If you don't like a particular colour, then simply change its hex value and this change will be reflected for all styles that use the colour. (Note that the AI doesn't necessarily use all defined colours).

This is followed by a section that defines colour combinations for foreground and background. Again shown below:

```
/* Theme colour combinations (Foreground / Background) */
.ow-theme-add5 { color: var(--sub5); background-color: var(--add5); }
.ow-theme-add4 { color: var(--sub5); background-color: var(--add4); }
.ow-theme-add3 { color: var(--sub5); background-color: var(--add3); }
```



```
.ow-theme-add2 { color: var(--sub5); background-color: var(--add2); }  
.ow-theme-add1 { color: var(--sub5); background-color: var(--add1); }  
.ow-theme      { color: var(--sub5); background-color: var(--theme);}  
.ow-theme-sub1 { color: var(--add5); background-color: var(--sub1); }  
.ow-theme-sub2 { color: var(--add5); background-color: var(--sub2); }  
.ow-theme-sub3 { color: var(--add5); background-color: var(--sub3); }  
.ow-theme-sub4 { color: var(--add5); background-color: var(--sub4); }  
.ow-theme-sub5 { color: var(--add5); background-color: var(--sub5); }
```

If you find that a foreground / background combination doesn't work for you, simply change its variable (`var(--xxxY)`), to one of the others. I have found that for some themes, I needed to use the lighter colour on the theme background while on others the dark colour was more suitable. Occasionally this needs to extend to backgrounds on either side of the main theme background.

It is also possible that you might want to modify the either combination colours for hovers.