

Android Notice Board App

Ensure your Android device allows installation of apps from unknown sources (see Settings | security).

Download the NoticeBoard.apk file from the IDRATEK site. Typically it will not install directly and you will have to use the native Android file manager on your device to locate the file in the Download folder. Tapping on this should then install it.

When the App installs for the first time it should take you straight to its settings page (you can return to this at any time from the main page by long clicking on the date field):

Site URL
User Name
Password
Target Notice Board Name
Refresh Period (s)
Text Size

Here you need to enter various parameters:

Site URL: This is the URL you would normally use to access your Cortex mobile server either via LAN or WAN. For example. <http://192.168.2.10:1087> might be the kind of structure for your LAN IP/port based URL.

User Name: In the Cortex web server object you need to add a new user whose purpose is to service Web API requests - as opposed to users for normal mobile browser access. For this new user you should enable the API access check box. See example below:

The screenshot shows the 'Web Server User' configuration window. It includes fields for 'User Name' (XYZ) and 'Password' (masked). Under 'Allowed Logon Times', 'On time' is 09:00:00 and 'Off time' is 20:00:00, with 'All the time' checked. 'Web Site Mode' has 'Enabled' and 'Allow control' checked. 'Allow logon when' has 'Not presence dependent' selected. 'Sender' has 'Enable Verification' unchecked and 'IP Address' set to 123.123.123.123. 'Web API' has 'Allow Access' checked. 'Style of web pages generated' has 'Style used (XSL file)' set to 'Cortex Mobile * Subject to licence' and 'Theme Identifier' set to 1.

You should not enable control or viewing access for any objects for this user. Only Web API access needs to be enabled for relevant objects, which can be done later via their Web server properties.

In this example the User Name would be entered into the Android App setting as XYZ

Password: Whatever you chose for the Cortex web server user XYZ

Target Notice Board Name: This will be the name of the target Notice Board object in your Cortex database which the App is to access. It might be an existing object whose content you wish to cast to the App, or you can add a new Notice Board object into your database specifically to service the App.

Refresh Period (s): The App polls Cortex every s seconds to obtain any data updates. A default of 10s is used but you can decrease or increase this value depending on requirement. Note that a single tap on the time field (see main app page) will always elicit an immediate request.

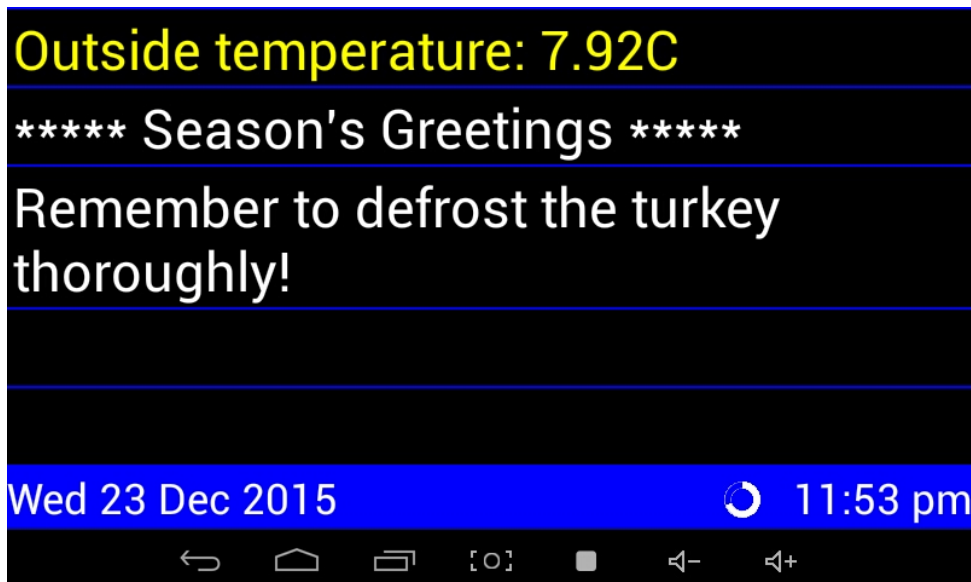
Text Size: Allows you to adjust the size of the text

Once the parameters are set up just hit the Android return button and this will then take you to the main display page. A long press on the date field (lower left) will take you back to the settings page.

A refresh period progress icon can be seen adjacent to the time field.

The date/time bar will flash blue/red if there is a problem connecting to the Cortex server. This will be cleared once the connection is corrected and when the next refresh period is up.

Note: the App is designed only to display the Message Area and the Notification Area contents of the Cortex Notice board object. Furthermore, notification messages are always displayed in yellow text at the top of the App page. A typical example is shown below:

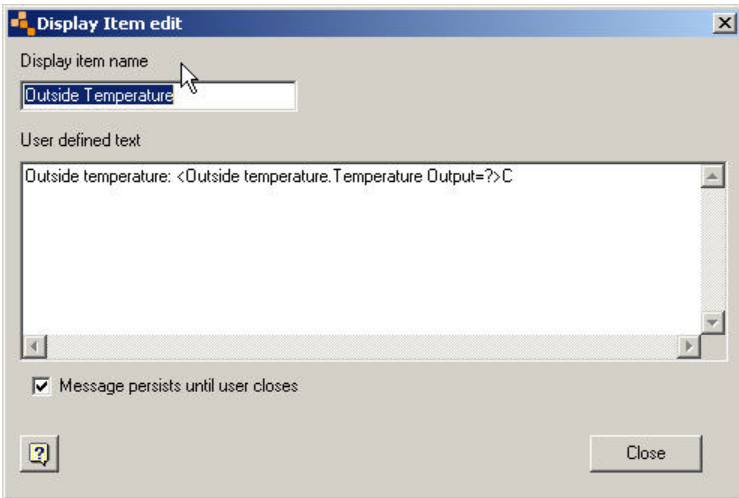


Notes about the Cortex Notice Board Object

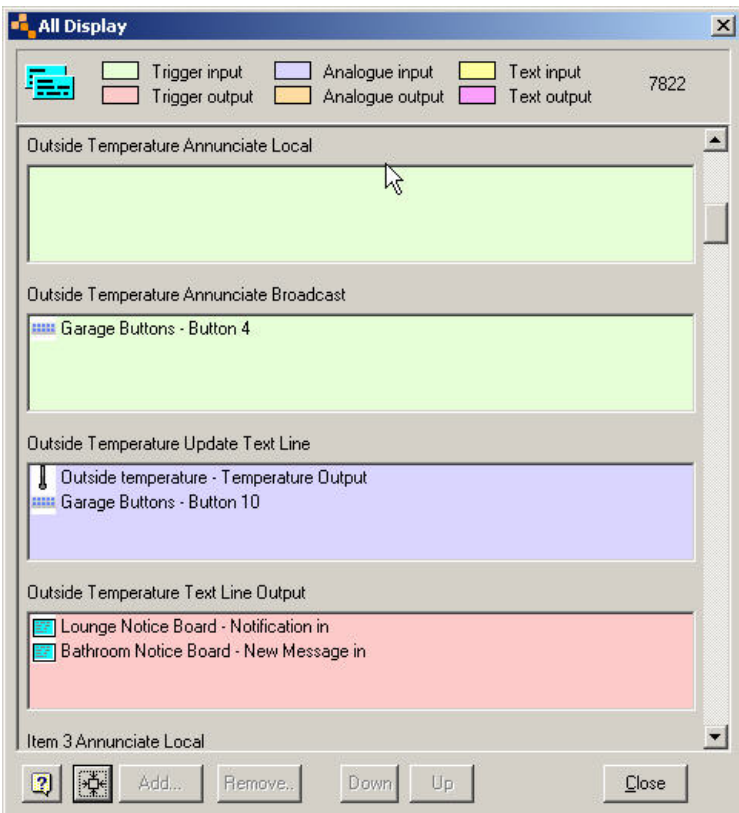
The Cortex notice board object is not limited to displaying direct text messages. In fact it is a rather more sophisticated beast which can be configured to display data values or states from objects and to use tags in order that fields within fixed text can vary. It can even change certain wording based on time/date to provide more natural annunciation. For example as an appointment draws nearer a date might become automatically reworded as 'tomorrow' or 'in 10 mniutes'. Multiple messages can be displayed. Some might be instigated by changes in data, others might be scheduled, and so on. With Cortex's Web API client feature you can even route messages from 3rd party servers – such as traffic data or weather reports (see Cortex help for details and examples).

How to route signal data items to a Cortex notice board object

Data items (for example a temperature value, a meter reading, etc.) are routed to the notice board via the All Display object (see Structure view under General branch). See Cortex help on the All Display object. An example item entry to display a temperature sensor output is shown below:



The output from this item must then be connected to the relevant notice board object. This is accomplished via the All Display object connections menu entries for this particular item. An example is shown below:



Note here that the Text Line Output of the item has been connected to two notice board objects, in one case as a notification message and for the other as a main message area message. Also notice that the Update Text Line field has two connections, one to the temperature output itself and the second to a button on a DFP module, Button 10 in this case. This ensures that the data to the Cortex notice board objects is updated whenever the temperature value changes OR if Button 10 on the DFP module is pressed. There is also an entry in the Annunciate Broadcast field which is not relevant to the notice board but illustrates multiple usages of the All Display object.

Web API Access Settings

Note: In order to access objects via the Web API you need to have the Web API feature enabled generally. This is accomplished via the Tools | Cortex API menu, in which the WebAPI Enabled option must be ticked. You must also enable access for the particular object you wish to expose (i.e the relevant Cortex notice board object in this case). To do this, right click on the object and navigate to its Web Server Options menu. Here the Web Enabled option must be ticked and also, for the user assigned to this object, the Viewing Allowed and Web API options must be ticked