roGridScreen

> **This component is deprecated.**
>
> **Beginning July 1st, 2017**, any new channels using this component will be rejected during certification.
>
> **Beginning January 1st, 2018**, any updates to existing channels using this component will be rejected during certification.

The Grid Screen provides a graphical display of poster art from multiple content categories from within a single screen.

Supported Interfaces

- ifGridScreen
- ifHttpAgent
- ifSetMessagePort
- ifGetMessagePort

Supported Events

- roGridScreenEvent

Description

Users can browse within a category list by scrolling horizonally and between category lists by scrolling vertically. There is an optional callout box in the lower right corner of the screen that can display details about the focused item without leaving the screen. Each item in the grid screen is represented by an image (aka poster), so any type of item that can be visually represented by an image can be displayed in the poster screen. It is used to show lists of data to users and common patterns include content categories, movies, podcasts, pictures, and search results.
The initial release of roGridScreen only enabled the default list style, "portrait", using the following art sizes:
Artwork sizes: SD=110x150; HD=210x270
It also required grid posters to be .jpg files.

Later firmware versions added mixed aspect ratio grids, and the ifGridScreen interface SetListPosterStyles() to set the aspect ratio of each row in the grid. If you want a mixed aspect ratio grid, you must call SetListPosterStyles() before you call SetContentList(), to avoid possible distortion of the graphic images in the grid.

Since Firmware version 2.8:

File types of .png and .gif files are now supported, though they are converted internally to .jpg by the roGridScreen so they have a performance penalty.
In v2.8, there are now multiple grid styles that are specified in the SetGridStyle()method below. It's also worth going back and reviewing the appManager theme parameters in roAppManager, as v2.8 adds some new grid parameters. The border around the focused poster screen can be customized with the GridScreenFocusBorder images in png format. PNG files can have a transparent color value that you will need to allow the focused poster image to show through the border image. The corresponding offsets should be negative offsets that would be up and to the left of the top left corner of the poster image. The width of the borders should be the absolute values of the offsets and the rest of the image should be transparent inside. The GridScreenDescriptionImage is also positioned relative to the top left corner of the focused image. It can be positioned up and to the left with negative x and y offsets, below and to the right with positive offsets, or in the other corners with mixed signed x and y offsets. It's recommended that you include a "callout" tip pointing to the focused image in the GridScreenDescriptionImage.

Diagram: roGridScreen

This object is created with no parameters:

- CreateObject("roGridScreen")

Example

```
Function Main()
 port = CreateObject("roMessagePort")
 grid = CreateObject("roGridScreen")
 grid.SetMessagePort(port)
 rowTitles = CreateObject("roArray", 10, true)
 for j = 0 to 10
  rowTitles.Push("[Row Title " + j.toStr() + " ] ")
 end for
 grid.SetupLists(rowTitles.Count())
 grid.SetListNames(rowTitles)
 for j = 0 to 10
 list = CreateObject("roArray", 10, true)
 for i = 0 to 10
             o = CreateObject("roAssociativeArray")
             o.ContentType = "episode"
             o.Title = "[Title" + i.toStr() + "]"
```

```
            o.ShortDescriptionLine1 = "[ShortDescriptionLine1]"
            o.ShortDescriptionLine2 = "[ShortDescriptionLine2]"
            o.Description = ""
            o.Description = "[Description] "
            o.Rating = "NR"
            o.StarRating = "75"
            o.ReleaseDate = "[<mm/dd/yyyy]"
            o.Length = 5400
            o.Actors = []
            o.Actors.Push("[Actor1]")
            o.Actors.Push("[Actor2]")
            o.Actors.Push("[Actor3]")
            o.Director = "[Director]"
            list.Push(o)
        end for
        grid.SetContentList(j, list)
    end for
    grid.Show()
    while true
        msg = wait(0, port)
        if type(msg) = "roGridScreenEvent" then
            if msg.isScreenClosed() then
                return -1
            elseif msg.isListItemFocused()
                print "Focused msg: ";msg.GetMessage();"row: ";msg.GetIndex();
                print " col: ";msg.GetData()
            elseif msg.isListItemSelected()
                print "Selected msg: ";msg.GetMessage();"row: ";msg.GetIndex();
                print " col: ";msg.GetData()
            endif
```

```
        endif
    end while
End Function
```