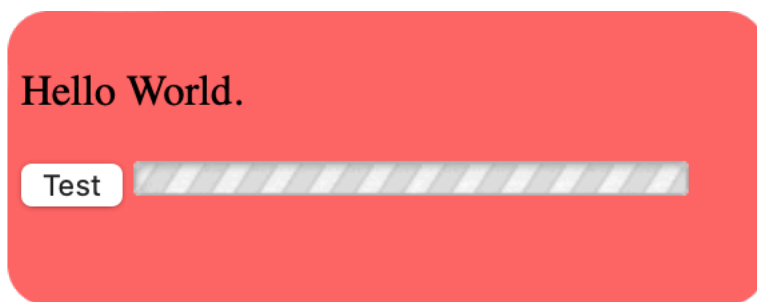


## Add a WebViewer to an Overlay window

Earlier this year we added the [Overlay](#) functions to our [MBS FileMaker Plugin](#). You can use them to show a graphic on screen outside or on-top of the FileMaker windows on macOS and Windows. This is great for showing splash screen with your logo when the solution starts, some navigation or arrows for a tutorial. What you show is up to you and some people just use it as a notification banner.

For version 14.5 we add a new function called [Overlay.AddWebViewer](#) to the plugin. Use this function to add a web viewer to the overlay instead of a picture. The web viewer can load some HTML plus CSS to show a webpage with transparency. This may include showing images, an animated GIF or even a movie. The website could also have some JavaScript to do some animation.



Let's start with a script to create an overlay with a web viewer. The [Overlay.AddWebViewer](#) function takes the overlay and adds a web viewer using WebKit on macOS or Microsoft's WebView2 control on Windows. Once we have it, we can load an URL with a HTML Page from our website and show the overlay. Loading the webpage from our website allows us to adjust it later.

*# First create an overlay*

```
Set Variable [ $overlay ; Value: MBS("Overlay.Create") ]
```

```
Set Variable [ $r ; Value: MBS("Overlay.SetFrame"; $overlay; 100; 100; 300; 120) ]
```

*# then add a web viewer to it*

```
Set Variable [ $$web ; Value: MBS("Overlay.AddWebViewer"; $overlay) ]
```

*# and load a website with transparent background*

```
# using <body class="background:rgba(255, 255, 255, 0.5)">
```

```
Set Variable [ $r ; Value: MBS("WebView.LoadURL"; $$web; "https://  
monkeybreadsoftware.com/filemaker/examples/transparent.html") ]
```

*# turn it visible*

```
Set Variable [ $r ; Value: MBS("Overlay.SetVisible"; $overlay; 1) ]
```

```
Set Variable [ $$overlay ; Value: $overlay ]
```

For the next example we build the HTML on the fly. We prepared a HTML template to use. This template includes the CSS keys needed to make the edges transparent. This includes declaring a transparent background for the body and

then a half transparent block on top to show the content. For the content we have two lines with different font sizes to show a "Please wait" message for a longly process and inform the user to wait. Here is the HTML use:

```
<html>

<head>
<meta charset="UTF-8">
<title>Hello</title>
</head>

<body class="background:rgba(255, 255, 255, 0)">
<div style="border-radius: 15px; background:rgba(100, 100, 100, 0.8);
padding:5px;">
<p style="text-align:center; font-size: 24pt; margin-block-start: 5pt; margin-block-
end: 5pt; ">Line1</p>
<p style="text-align:center; font-size: 14pt; margin-block-start: 5pt; margin-block-
end: 5pt; ">Line2</p>

</div>
</body>

</html>
```

Next we need the script to position our webviewer based banner. We start with the text to use, hard coded for now, but you could change that to a script parameter. Then we calculate size of screen. Our [Screen.Width](#) function returns pixels on Windows and points on macOS. Maybe in the future we can change that. For now we calculate the center of the screen for the frame of our new overlay.

Next we take the HTML template and replace the placeholders with our values. Don't forget [Text.EncodeToHTML](#) to encode the HTML characters. Then we can use [WebView.LoadHTML](#) to load the content and turn the overlay visible. Since it shows a banner as a notification, we free it after 5 seconds. Here is the sample script from our Overlay.fmp12 sample file:

```
# the text to show
Set Variable [ $line1 ; Value: "Exporting" ]
Set Variable [ $line2 ; Value: "Please wait while we export records" ]
#
# check screen size
Set Variable [ $ScreenWidth ; Value: MBS( "Screen.Width" ; 0 ) ]
Set Variable [ $ScreenHeight ; Value: MBS( "Screen.Height" ; 0 ) ]
If [ MBS("IsWindows") ]
    Set Variable [ $scale ; Value: MBS( "Screen.Scale" ; 0 ) ]
```

```

        Set Variable [ $ScreenWidth ; Value: $ScreenWidth / $scale ]
        Set Variable [ $ScreenHeight ; Value: $ScreenHeight / $scale ]
End If
#
# First create an overlay
Set Variable [ $overlay ; Value: MBS("Overlay.Create") ]
Set Variable [ $r ; Value: MBS("Overlay.SetFrame"; $overlay; $ScreenWidth / 2 -
200; $ScreenHeight / 2 - 100; 400; 120) ]
#
# then add a web viewer to it
Set Variable [ $$web ; Value: MBS("Overlay.AddWebViewer"; $overlay) ]
#
# Build HTML with our texts
# and load a website with transparent background
# using <body style="background:rgba(255, 255, 255, 0.8)"> for a transparent
background
Set Variable [ $html ; Value: "<html> <head> <meta charset=\"UTF-8\">
<title>Hello</title> </head> <body class=\"background:rgba(255, 255, 255, 0.8)\">
<div style=\"border-radius: 15px; background:rgba(100, 100, 100, 0.5);
padding:5px;\"> <p style=\"text-align:center; font-size: 24...\" ]
Set Variable [ $html ; Value: Substitute($html; "Line1";
MBS( "Text.EncodeToHTML";$Line1)) ]
Set Variable [ $html ; Value: Substitute($html; "Line2";
MBS( "Text.EncodeToHTML";$Line2)) ]
Set Variable [ $r ; Value: MBS("WebView.LoadHTML"; $$web; $html) ]
#
# turn it visible
Set Variable [ $r ; Value: MBS("Overlay.SetVisible"; $overlay; 1) ]
Set Variable [ $$overlay ; Value: $overlay ]
# and auto hide after 5 seconds
Set Variable [ $r ; Value: MBS("Schedule.EvaluateAfterDelay"; 5;
"MBS(\"Overlay.Release\"; " & $overlay & ")") ]

```

Please try and let us know what you think about it. The combination of Overlay and Webviewer functions gives a ton of possibilities. In future we may build on top of this. But for now you can improve with using CSS animations and JavaScript in the web page you load.