

## AOZ Studio Beta - Bug #547

### Suggested modification to rfonts.js to overcome the 'cannot\_load\_font' issue

10/13/2020 04:06 PM - Phil Bell

<b>Status:</b>	New	<b>Start date:</b>	10/13/2020
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0:00 hour
<b>Target version:</b>			
<b>Affected version:</b>	Beta RC3		
<b>Description</b>			
<p>Through lots of testing on mobile it is often that the <code>Fonts.prototype.loadFont</code> function will fail to load a font in the limited 3000ms timeout, this is not really a problem as the font will eventually be loaded by the browser and be available to AOZ.</p> <p>This behaviour explains why we have seen the alert popup on first load but then a refresh will successfully load the application.</p> <p>Currently, if the timeout occurs the <code>Fonts.prototype.loadFont</code> function will set the 'cannot_load_font' error, the user will see a browser alert with this error message and be left with an unresponsive "blank" screen.</p> <p>I suggest the <code>Fonts.prototype.loadFont</code> should be modified to ignore the timeout error and allow the app to continue.</p> <p>Below is the modified function, I have tested on mobile and using the network limiter in the Chrome Dev Tools</p> <pre>Fonts.prototype.loadFont = function( fontInformation, weight, italic, stretch, callback, extra ) {   weight = typeof weight == 'undefined' ? 'normal' : weight;   italic = typeof italic == 'undefined' ? 'normal' : italic;   stretch = typeof stretch == 'undefined' ? 'normal' : stretch;    var self = this;   if ( fontInformation.type == 'google' )   {     var fontStyleIndex = 'italic:' + italic + 'weight:' + weight + 'stretch:' + stretch;     if ( !fontInformation.fonts[ fontStyleIndex ] )     {       var font =       {         weight: weight,         style: italic,         stretch: stretch       };       var observer = new FontFaceObserver( fontInformation.name, font );       this.aoz.loadingMax++;       observer.load().then( function()       {         self.aoz.loadingCount++;         font.fontInformation = fontInformation;         font.baseLines = [];         fontInformation.fonts[ fontStyleIndex ] = font;         callback( true, font, extra );       }, function(e)       {         // Phil Bell 13/10/2020         // Dont error if timeout exceeded, maybe slow network connection, font will load eventually         if (e.message.indexOf('timeout exceeded') &gt; -1)         {           self.aoz.loadingCount++;           font.fontInformation = fontInformation;           font.baseLines = [];           fontInformation.fonts[ fontStyleIndex ] = font;           callback( true, font, extra );         }       }     )   } }</pre>			

```

    }
    else
    {
        self.aoz.loadingCount++;
        self.aoz.loadingError = 'cannot_load_font';
        callback( false, null, extra );
        callback( true, fontInformation.fonts[ fontStyleIndex ], extra );
    }
    } );
}
else
{
    callback( true, fontInformation.fonts[ fontStyleIndex ], extra );
}
}
else if ( fontInformation.type == 'amiga' )
{
    if ( !fontInformation.font )
    {
        var path = './resources/fonts/amiga/' + fontInformation.name + '.js';
        this.aoz.loadingMax++;
        this.utilities.loadScript( path, {}, function( response, data, extra )
        {
            self.aoz.loadingCount++;
            if ( response )
            {
                fontInformation.font =
                {
                    fontCode: AmigaFonts[ fontInformation.name ],
                    fontInformation: fontInformation,
                    canvasDefinitions: [],
                    baseLines: [],
                    italic: italic,
                    weight: weight,
                    stretch: stretch
                };
                callback( true, fontInformation.font, extra );
            }
            else
            {
                self.aoz.loadingError = 'cannot_load_font';
                callback( false, null, extra );
            }
        } );
    }
    else
    {
        callback( true, fontInformation.font, extra );
    }
}
};

```