

# Performance FAQ

## Table of Contents

- Can I have a background video playing in my channel navigation screen?
- Will using less task nodes increase my performance?
- How many animations can I use?
- How can I view system memory?
- What is the difference in animating different resolution images?
- How many items can I put in a RowList / MarkupGrid / PosterGrid, etc.?
- Help! My channel is still lagging!
- Can persistent data be stored across different channel sessions?

To find exactly what you are looking for, you may use your browser "FIND" button to locate an item on this page.

Question	Answer
<b>Can I have a background video playing in my channel navigation screen?</b>	Yes. A common channel design is to have a video playing in the background with a RowList or GridList menu overlaid on top. Check out an example implementation <a href="#">here</a> to test the performance implications of such a design. There are delays while the video is loading (it is therefore recommended to use a low-res video for something unimportant like a background video), but once it is loaded, there are no further performance decreases that can be attributed to the inclusion of the video.
<b>Will using less task nodes increase my performance?</b>	Task nodes are an essential tool for a channel looking to make any asynchronous calls separate from the main BrightScript thread and the render thread. At the same time, spawning too many different threads is an easy way to create laggy and unresponsive channels. Not only should you take care to follow all of the guidelines listed in the <a href="#">Task node class documentation</a> , be aware that a common problem is calling too many simultaneous URL requests from multiple Task nodes. Depending on network speed and other factors, this could cause serious performance issues and result in HTTP timeouts which prevent content from reaching your users.
<b>How many animations can I use?</b>	<p>Testing on a higher end device with a dedicated GPU like a Roku 3 has shown it can handle up to 800 <a href="#">Animation Nodes</a>, each with its own Interpolator, running at the same time without any performance issues. This is likely far more than a reasonable channel will ever need to use. However, on a Giga, the channel showed visible lag at just 50 simultaneous Animation Nodes. Even worse was on a Tyler, which had trouble with 20 Animation Nodes at the same time. These tests were run on a channel with no other components, so take the exact numbers with a grain of salt. In a real channel, there will likely be other things (Task threads, loading content, user channel navigation, etc.) that will compete for system resources. Instead, you should take note of the large difference between the performance capabilities of a high end device and a low end device and remember to develop your channel accordingly.</p> <p><b>Use Animations Somewhat Sparingly.</b> Animations and their associated interpolators require a non-inconsequential amount of computational power (some quantitative data can be found in the FAQ below). An excess of Animation nodes running simultaneously can cause performance issues, although non-concurrent animations will be no problem. <i>However, keep in mind it is generally not an excess of animations that cause performance issues (although it is certainly possible).</i> Rather, poor texture management or interference from other threads are usually the root cause of stuttering animations. Having many animations can exacerbate any performance issues a channel is having and make it exceedingly obvious to the user. In that case, it would be best to simply avoid animated transitions for large UI layouts with multiple complex elements.</p>

<p><b>How can I view system memory?</b></p>	<p><b>For SceneGraph apps</b></p> <p>For SceneGraph apps, just like above, run the channel and <b>port 8080</b>. Then run "sgnodes all"</p> <p><b>For pre-SceneGraph apps</b></p> <p>For pre-SceneGraph apps, telnet to the console; hit <b>^C</b> to break into the debugger; run "bcs" or "bcs"</p> <p><b>Note:</b> This is also useful for SceneGraph apps.</p>
<p><b>What is the difference in animating different resolution images?</b></p>	<p>There was no noticeable performance difference between animating a full screen, high resolution image and animating a small Rectangle. While there may be other costs involved, like in the load time / render time / texture memory usage of the image, the actual CPU usage of the Animation and Interpolator did not change depending on what kind of content it was applied to.</p>
<p><b>How many items can I put in a RowList / MarkupGrid / PosterGrid, etc.?</b></p>	<p>In theory, as many as you want. SceneGraph Lists and Grids utilize a content manager that automatically loads and unloads bitmaps from texture memory as items in the list/grid are revealed and hidden. Try out this <a href="#">test channel</a> and notice that once the RowList is loaded, having hundreds of images in each row shows little to no performance difference over having just ten - even on a low end device. However, keep in mind that the initial loading of all of the content into the list or grid still takes time, so trying to download too many large assets or data at once will result in delays and significant load times.</p>
<p><b>Help! My channel is still lagging!</b></p>	<p>Go through this checklist one more time.</p> <ul style="list-style-type: none"> <li>• Switch all your images with the lowest resolution possible (before a noticeable decrease in visual quality). A 3840x2160 image and a 360x240 image will look exactly the same on a 360x240 Poster.</li> <li>• Avoid using or gracefully degrade large scale UI animated transitions for lower-end devices. Not that it takes up an especially large amount of resources, but a full-screen image fading in and out in the background makes it blatantly obvious when your channel is experiencing even the slightest performance lag.</li> <li>• Reduce the amount of content loading you're doing at once. If you have multiple Task nodes running hundreds of urlTransfers at once, there will be lag. Load content dynamically as the user needs it.</li> <li>• Change the channel's UI to reduce the number of rendered nodes on the screen. If you want 50 Poster nodes overlaid on a changing background video with an animated HUD, then it's likely time to reconsider the UI and compromise for low-end hardware compatibility.</li> </ul>
<p><b>Can persistent data be stored across different channel sessions?</b></p>	<p>Yes, developers can use the caches: file system to allow applications to cache data to volatile or persistent storage instead of tmp:. End-users can add an external SD card to their device which will preserve the data even after a system reboot and improve performance. Users without extended storage also benefit from the use of a shared in-memory cache that is automatically managed by the system to optimize for the most recently used assets.</p> <p>Note that the OS can evict the data at any time, such as when another channel decides to write so much data that space is required. Thus, a channel should always check for the existence of the file they wrote to before relying on the data cache. Please note that this is currently a Beta feature.</p>