The eXastum Web Desktop

Brian Millar

Contents

0.1 Preface

Welcome to the eXastum project! Thank you for taking an interest and especially thanks to you for actually taking the time to read the documentation. This document is designed to serve both users and developers. I hope you will find this document useful in using eXastum and hopefully building great

HTML5 apps that integrate with the eXastum web desktop. This document is not very large, the eXastum source code itself is pretty short and easy to understand, such a simple project would not be well served by a large manual. Have fun and if you have a question, need help or see a better way of doing something then please feel free to get in contact with me. To find my contact info or any other information you seek please visit ontropy.org.

0.2 The eXastum Design

I feel this quick introduction is needed to explain something many of you will find strange about the eXastum source code. I generally just write it the way that makes the most sense to me and the way that just works. I don't follow a super strict design or style in eXastum. My approach is generally a C-like one with some functional aspects. I generally use functions for everything and OOP is not really used. The only times I use objects is when I have to to access the DOM or get a date or something. eXastum defines no objects of its own. I feel I need to say this to prevent anyone sending patches or mail trying to make the whole thing some pure object oriented system. I don't support OOP but I do believe in modular design. Each part of the system gets its own file. That is important.

0.3 Basics

eXastum itself is a fairly basic system and is not hard to understand. Your app is run in an iframe with a decorative window wrapped around it. Some metadata provided by your app tells eXastum how to run it.

0.3.1 Calling eXastum Functions

You can call any eXastum system function using parent.functionNameHere(); the parent part is important as your app runs in an iframe under eXastum its namespace is different, to call functions seen by eXastum you must 'jump' up a level and then call it.

0.4 The Application Manifest

TODO here

0.5 Including Libraries

As normal any app is free to include its own libraries as it normally would from the web or its own folders however for popular libraries eXastum includes a loading feature that allows the user to have only one copy of a library for all applications. An example of one included with eXastum is Three.js (see "Using 3D Graphics"). You can call parent.requestLib(libname) to ask eXastum to load the library you desire for you.

0.6 Using 3D Graphics

eXastum itself does not provide any functions or features for dealing with 3D graphics, it does however include the Three.js library for creating webGL content. The system uses this library for things like the audio visualizers. You can include it in your apps as well, this saves the user from having several copies of the same library taking up space on their disk. To do this see the section called "Including Libraries"

0.7 The Audio System

eXastum uses the Web Audio API to allow interesting audio features such as visualization.

0.7.1 Connecting to eXastum's Audio System

To have the system-wide audio features (such as system volume control) affect your app you need to tell eXastum about your audio sources. This is extremely easy to do. Calling eXastum's addAudioSource() function on your audio object will add it to the audio sources the system can see. That's all that is needed. You must call that on each audio object. If you have an audio tag in your HTML you can add it using it's ID, like so: parent.addAudioSource(document.getElementById("myAudioTag")); The 'parent' part of the call is explained in this document in the section called "Basics".