Cedar

RSL Seeder and Download Monitor

Introduction

Cedar fulfills two functions. Its primary purpose is to enable the silent seeding of RSLs to a client machine in advance of the deployment of an application using those RSLs. By seeding the RSLs in advance, any flashplayer that encounters the seeder will cache any signed RSLs and any browser will cache any non-signed RSLs. Obviously this will mean that when the player/browser encounters the live application at a later point it, load time will be greatly improved as the RSLs it uses will already be cached to the client machine. Secondly Cedar allows monitoring of RSL loading including individual load times for each RSL.

Given that a new Signed RSL is needed for every update to that RSL's codebase, and that the combined size of RSLs for the Flex 4 SDK (Including the OSMF and Text Layout RSLs) is non-trivial, this approach allows a more effective solution to pushing out new RSLs *with* your application, as it will minimise the disruption to the client caused by exessive load-times.

Obviously this approach will only be effective if the client has encountered the seeder previously to having encountered the application, therefore it will be effective only in situations where there is a regularity of visitor and a long enough lead-in time to allow enough regular visitors to have encountered the seeder before encountering the application.

Usage

Cedar consists of two SWFs. The Seeder and the Seed. The Seeder sits embedded in the wrapper and loads the Seed. The Seed in turn loads the RSLs.

Important: Unfortunately you will probably need/want to pay a visit to the shameful Flash Player Settings Manager to turn of your player's ability to cache signed RSLs. If you do not do this, the seeder will use the RSLs that are probably already cached in your player, or will only download them a single time.

Flash Player's *Website Storage Settings Panel* is accessed here (If you get a script timeout, refresh the page):

http://macromedia.com/support/documentation/en/flashplayer/help/settings_manager07.html#117717.

Untick the check-box *Store Common Flash Components To Reduce Download Times.* Once unchecked, I suggest you go to a different panel, then return to make sure your selection stuck. You may or may not need to restart your browser.

The Seed is a simple swf that has been compiled to use RSLs. It's sole purpose is to trigger the download of any signed and unsigned RSLs that you want to be cached. The Seed currently in the codebase uses all the RSLs included with the Flex 4.0.0.14159 SDK.

The Seed uses a custom preloader to forward rellevent events out in the form of bubbling events. These events are fundamental to the ability of the Seeder to monitor the RSL load, though obviously the RSLs will still load without them. There is nothing else special about the seed other than the fact it is compiled using RSLs and that it dispatches these events. If desired, you can use any Flex Application as the Seed, using the custom preloader for your own custom preloader to dispatch the events needed by the Seeder or dispatching the same events from your own preloader.

The Seeder is an AS3-only SWF that loads the Seed. It can be set to begin loading after a specified amount of time, or as a result of a call to its load() function from the wrapper via ExternalInterface. These options allow you control over the point at which you want the RSLs to load, so that you can be sure all the page content has fully loaded before using the client's bandwidth on the RSL load.

The Seeder can be run in debug mode, which means it will display the RSLs' status, including how long each RSL has taken to load. *Important note: When the Seeder is not in debug mode it hides all UI.* To set it's mode see the Flashvars below. In debug mode the Seeder displays load progress for each RSL and total load progress across all the RSLs. It also displays the loaded swf below the text display.

You can also specify a Javascript method in the wrapper to handle status updates. If a method name is provided, the Seeder will call that method whenever it has something to report, passing the latest update as an argument. How you impliment this method is up to you.

Configuration

The Seeder is configured by the following Flashvars:

swf_path
Path to the Seed swf. Obviously if you are loading the swf from a different domain, the usual Security rules apply.

debug true|false (default:true)
Should the Seeder display debug info and pass it out to the wrapper if 'wrapper_update_function_name' is defined. If false, hide all UI/Display

use_timeout (default:false)
If useTimeout is set to false, we wait until load() is called on the SWF by the wrapper via ExternalInterface. Otherwise we wait for *timeout_duration*, then load it.

timeout_duration (default:3000)
Duration *in 1000ths of a second to wait before loading the Seed*

wrapper_update_function_name
Name of the Javascript function in the wrapper handling status updates. Leave blank if you don't want the swf to push updates to the wrapper.

ANT

Included in the project are a set of ANT build files to make compilation of the Seed with RSLs easier (Alongside compilation of the Seeder). You will need to customise a number of the property files as needed for your own system (Including the loaction of your chosen Flex SDK), but the property file that defines the RSLs is *build/ant/properties/seed.properties*. It is heavily commented and should make sense to anyone with a knowledge of ANT and MXMLC.

Supporting Documentation

Flash Player Settings Manager:
http://macromedia.com/support/documentation/en/flashplayer/help/settings_manager.html

RSLs
http://www.adobe.com/devnet/flex/articles/flash_player_cache.html

RSLs in Flex
http://help.adobe.com/en_US/flex/using/WS2db454920e96a9e51e63e3d11c0bf69084-7add.html