MySpace DataRelay

What is Data Relay?

Data Relay is MySpace's middle tier. At it's core, it is a messaging system for plug-in based servers. Included in the release of Data Relay are three of our major plugins:

* Forwarder - This handles the actual moving of messages, both from client to server and between servers
* BerkeleyDB - This handles storing data, and is the component used for basic key/value caching.
* Index Cache - This is a two tiered indexed data system, for more advanced queries than simple key/value.

The release also includes our network transport and serialization system, as they are required to run at MySpace's scale, and we wanted to release exactly what we use. The folder structure of the solution mirrors our internal source control, which is why there are more folders than seem strictly necessary.

Getting Started

Currently DataRelay is only available in source form - please download "Latest Version" from the Source Code tab. DataRelay relies on Microsoft's CCR library, the installer for which can be found at "/Prerequisites/Microsoft CCR and DSS Runtime 2008 R2 Redistributable.exe". Please run this so that you have the necessary dll available.  
  
After doing so, open the Solution DataRelay-OpenSource.sln at the root, select your platform (there is unmanaged code that needs to know if you are running on x86 or x64), and build the solution. Please note that "AnyCPU", "x86", and "win32" are equivalent. Some of our projects currently use different platform naming conventions and we have not unified them yet.  
  
Data Relay is a windows service on the server side, and by default the windows service and components build into "\Infrastructure\DataRelay\DataRelay.WindowsService\Output". In that folder there is a powershell script called "InstallService.ps1" and a regular dos batch file called "InstallService.cmd" which will install the service at that location. Once the service is installed, you can start it from your window services snap-in - by default it is named, unsurprisingly, "MySpace DataRelay".  
  
Starting the service will produce a fair amount of information in "\Infrastructure\DataRelay\DataRelay.WindowsService\Output\RelayNode.log". There should hopefully be no problem starting the service, but any issues will be reported here.  
  
The default configurations will load BerkeleyDb as an in-memory only store for 100 numeric ids, which allows you to store 100 segregated sets of objects.  
  
There is a simple example client that can perform gets, sets, and deletes of one object that stores string key to string value pairs in $/DataRelay/Infrastructure/DataRelay/DataRelay.SimpleConsoleClient. Running the output of that project will allow you to ensure that your local installation is running, and the code is an excellent place to begin exploring that DataRelay api.   
  
Regrettably there is currently very little API documentation, but we will be updating this space frequently with more information!

Wow That's a Lot Of Configuration

Yes. Yes it is. Data Relay is very flexible, but this comes with the downside of being rather complex. Here's a breakdown of the configuration files

* MySpace.DataRelay.WindowsService.exe.config
  + Controls the few settings that apply to the windows service itself. The real meat of DataRelay is loaded into a seperate app-domain, similar to how IIS operates. This allows the service to detect assembly updates and reload itself as needed.
* Logging.RelayService.config
  + The log4net settings for the windows service.
* Logging.RelayNode.config
  + The log4net settings for the actual message handling system.

The rest of the configuration is in the ConfigurationFiles\ folder of the windows service. Many of them are also used by the client.

* ConfigurationFiles\RelayNode.config
  + Controls base settings for the component host, and has pointers to the rest of the configuration files. Because some of the settings can be considered universal and some depend on the server role, the configuration is broken up into functional blocks.
* ConfigurationFiles\RelayComponents.config
  + Controls what components the server should be loading. Every components has some of the same settings - which class represents the component and what sort of messages should be delivered to it. In addition, every component has its own configuration, which it can define.
* ConfigurationFiles\RelayTypeSettings.config
  + Each data type stored in data relay has a few of its own settings - most importantly giving a numeric Id to the type and which group of servers the type is mapped to. Those settings are here.
* ConfigurationFiles\RelayNodeMapping.config
  + Each data relay client needs to know about the available servers, and each server needs to know where it fits in the overall scheme. This file contains the breakdown of each server, and some settings that are adjustable on a per-group basis. The default configuration contains only a single node, "localhost". as it assumes you will be running your first node on the machine that builds it.
* ConfigurationFiles\RelayTransportSettings.config
  + This controls which port the main socket transport listens on, and enables activation of the http transport for a (very) limited subset of operations.
* ConfigurationFiles\SocketClient.config and AsyncSocketClient.config
  + These control aspects of the socket transport client, some of which can be overridden on a per-group basis in the RelayNodeMapping configuration file.
* ConfigurationFiles\SocketServer.config
  + Controls aspects of the socket transport server.