Neptune: New Framework for the New Year

Posted At : January 3, 2011 1:30 PM | Posted By : Steve
Related Categories: Frameworks, Neptune

Many years ago, I was copying an application from one site to another. The task seemed simple enough, but it was taking a long time. First I had to copy the code. Then I had to create the tables/columns/primary keys. Then I had to create any folders being used to store uploaded file (or if they were part of the application, delete any uploaded files from the folders that I copied). Then I had to change the look of the program to match the site to which I was copying it.

I remember thinking that it should be easier. I resolved that I could make a "Ten Minute Install". The goal turned out to be a bit harder than I imagined at the time (though not nearly ambitious enough in the long run).

Years later, however, this would be a major piece of my framework. Except instead of a "ten minute install" I have about a ten second install. To copy a Neptune program from one site to another just requires copying the folder that holds the program and then browsing to it.

By doing that, the files are copied and the tables/columns/primary keys (and optionally some data) are created. It will also set up folders for uploaded files and run any custom installation code. The program will automatically use any settings (including appearance) on the new site.

In fair disclosure, this isn't my first post about this framework. I posted about it under the name "AFF" **about seven months ago**. In that time, the framework got a new name and has matured a bit as well. I have worked a lot on this framework in the last year.

What is Different and Why Should I use it?

Neptune is the easiest framework for most web development. If you have a large (non-framework) code base that you are thinking about migrating to a framework then Neptune is absolutely the easiest framework to migrate to. It isn't **hub and spoke** (where all calls go through index.cfm), so you just need to copy some files into your site and add a couple of lines of code and you are using the framework. You can take advantage of framework features as you need to - no need to change any code until you want to take advantage of the framework.

Neptune is the easiest framework to learn and use. You don't have to learn about "Object-Oriented programming" (though you can use OO with the framework) or "Inversion of Control" (which it does for you) or "Implicit Invocation", but it does automate most of the mundane tasks of programming. In fact, the framework itself handles much of what a code generator would normally handle - without needing to generate actual code.

Since Neptune uses services instead of objects, you can use familiar (and easy) recordsets instead of more complicated objects and object collections. But with the power and simplicity of **DataMgr**, you will have a data access layer solution that is **better than ColdFusion ORM**.

Rather than being about some theory about the framework itself (like "**small framework**" or "**implicit invocation**"), Neptune is geared towards making development life as easy as possible and making *your code* as concise and readable as possible. The development of Neptune has taken a very evolutionary approach. I have simply attacked the most tedious aspect of my development life again and again.

Neptune is largely driven by a collection of other projects that can each be used independently in addition to the core framework which ties them all together and enabled auto-installation of programs.

- **com.sebtools** package
- **SebTags** custom tag collection
- **Layout Components**
- **Page Controllers**

How do I get Started?

Even though this is just the first beta, Neptune already has quite a few resources to help you get started.

The **documentation** is still in development (currently: 36 separate pages, 43 printed), but should be helpful. I would recommend "**Getting Started with Neptune**" (based on **Dan Wilson**'s brilliant "**So you want to build a ModelGlue:Unity application?**" series).

The **code generator** isn't needed to quickly make Neptune programs, but it does speed up the process slightly and it is a great way to get up to speed for new users of Neptune. Define a data structure in the generator and see what code Neptune could use to manage it.

Over the next few weeks and months, I plan to release some **Jedi-quality programs** that will be useful in their own right but that will also serve as real-world examples of Neptune in use.

Final Thoughts

A lot of work has gone into Neptune already, but a lot more is still to be done. I would love any help I can get - finding bugs, writing code, or coming up with new feature ideas.

Neptune is open source and free for any use.