



White Paper

**APPLYING THE OBJECT-ORIENTED PHILOSOPHY
TO CONTENT MANAGEMENT AND WEB DEVELOPMENT**



Table of Contents

Executive Summary	3
The Object Oriented Philosophy	3
The Object Oriented Developer	3
Utilizing SR2iteRefresh to Apply Objected-Oriented Web Development....	5
The business benefit when utilizing Object-Oriented Programming	6



Executive Summary

As an organization matures, the architectural foundation it is built upon must also be able to mature. Words like scalability, flexibility, extendibility, and dependability are daily reminders of how critical our infrastructure choices are to the success of the eEnterprise. Whereas these words are the basis for all decisions regarding the architecture backbone on which the organization runs, organizations often succumb to monolithic, proprietary systems for specialized needs. More often than not, this has been the case in the Content Management System arena.

Today's organizations demand choice. Moreover, they demand best of breed solutions that function within their existing information architectures. Refresh Software has applied the principles of Object Oriented programming to the problem of Content Management and in doing so has created a standards based, loosely coupled Content Management solution that demonstrates the increases in freedom, flexibility, scalability, and reliability that this architectural pattern provides. This solution is [SR2SiteRefresh](#), offering component content management for the Content Driven Enterprise.

The Object Oriented Philosophy

The idea behind object-oriented programming (OOP) is that a computer program may be seen as comprising a collection of individual units, or objects, that act on each other, as opposed to a traditional view in which a program may be seen as a collection of actions, or simply as a list of instructions to the computer. Each object is capable of receiving messages, processing data, and sending messages to other objects. Each object can be viewed as an independent entity with a distinct responsibility.

One of the advantages of object-oriented programming techniques over linear programming techniques is that they enable programmers to create modules that do not need to be changed when a new type of object is added. A programmer can simply create a new object that inherits many of its features from existing objects. This makes object-oriented programs easier to modify.

The Object Oriented Developer

In the early stages of Web development, programming was all about in-line scripting. This was no surprise, as websites were basic HTML pages with few, if any, programming interactions. Websites today are much more complex, and in many cases serve as the front-end of a web-enabled application. In order to fully leverage these web-enabled applications, websites are now asked to connect to various types of databases and execute many programming interactions, all while offering strong performance metrics. As a result, these web-enabled applications are usually created by a team of developers, each with a specialized



skillset that is brought to bear on the project. With an object oriented approach, each specialized skillset can be applied to a very specific piece of functionality critical to the program. Resources are maximized, while delivering a robust solution in a timely manner.

The advantage of using [the SR2Refresh Software's SiteRefresh](#) solution as a content management system is that not only does it support .Net and JSP, but any scripting language may be employed to meet the desired goal. This open approach aids in the transition into [SR2iteRefresh](#) in that any developer that is just learning .Net or JSP can leverage their existing code to run the system, while slowly implementing an OO architecture. This is a huge advantage for businesses looking to implement OO programming but don't have the immediate resources to hire or train their existing staff.

Additionally, [SR2iteRefresh](#) offers an OOP API to provide a new level of simplicity to the transition to an object oriented model. Refresh Software also offers plenty of examples to get you jump started into OOP using .NET or JSP. By leveraging these examples, many developers immediately see the advantages of an OO programming model.

[SR2iteRefresh](#) delivers the following benefits to developers:

- Plenty of sample code and examples on how to implement SiteRefresh into your existing environment
- Ability for developers to share code
- Simplified development code
- A knowledgeable support staff that can assist you
- Developer training

If you're a Senior OO developer, you'll appreciate how easy it is to implement the [SR2iteRefresh](#) API into your own environment. When implementing a middleware application into a mission critical environment, the last thing a developer should be concerned with is restrictions and lost functionality. [SR2SiteRefresh](#) addresses this issue with the easy-to-implement [SR2iteRefresh](#) API, which is open source and completely configurable by developers.

[SR2iteRefresh](#) delivers the following benefits to Senior OO programmers:

- Access to the SiteRefresh Content Retrieval API source file
- Custom Developer training
- API Documentation
- Similar functionality between .Net and Java.

Utilizing **SR2iteRefresh** to Apply Objected-Oriented Web Development

At the heart of SiteRefresh is the abstraction of the content layer based on the OO model. Essentially this results in any page now potentially being viewed as several or many content "chunks" or content assets that makes up that page. This is a huge departure from the way most CMS approach page management today. By dividing the page up into its represented pieces or content asset objects we can now give them all special attributes. This is how we are now able to take advantage of one of the core concepts of the OO paradigm which is "Polymorphism", every content asset can be consumed at a minimum by using their Core Content Asset interface (common fields).

SR2iteRefresh applies the concept of defining content through asset types. An asset type is analogous to an OOP class and allows you to define attributes or data fields for each content item. A content asset is similar to an instance of an object of that class. In order to incorporate this OOP type of model into an "enterprise content context" the current candidate objects and their associated relationships and work flows must be exposed and organized through the application of information architecture (IA). Information architecture creates the foundation that allows for further decomposition of the content into objects. It is in the analysis of those objects that patterns emerge to help us define AssetTypes.

The OO approach to enterprise content essentially abstracts the content layer from the presentation format. This promotes the ability for easy content reuse across multiple "views" of the content. For example, a press release content asset may appear in the full text version on a page for presenting press releases while just the title and teaser appears on the home page as a listing of recent press releases. So in this example, the title and teaser may be used in one part of a web site (home page under recent announcements) and the whole body of the document can be used in two other places (the product section and the new release section). This one-to-many relationship for consumption is a key factor in managing and organizing your content for adroit use.

Additionally, the abstraction of the content layer also allows for a clean "separation of concern" between the authoring process and the developing process, i.e. content authors and web developers. This decoupling the content authoring process from the development process frees both groups from the needless corporate interdependencies and in other cases unnecessary entanglements that exist in most corporations today. In other words, the 'template development' responsibilities can be delegated to engineering while 'content development' is the purview of marketing and product stakeholders. This clean "separation of concern" allows both lifecycles (web page template, and content asset) to progress independent of each other.



This separation is only possible through a well understood definition of how the page template and the content object will interact when brought together (defining role and responsibility of the Asset Type). The asset type definition constructs the relationship and the rules of interaction between the content object and the template.

The business benefit when utilizing Object-Oriented Programming

The main advantage of utilizing an object-oriented programming technique over a linear programming technique is that OOP enables programmers to segment programs into modular objects such that when a new module is created, the pre-existing modules do not require modification. Additionally, programmers can simply create a new object, based on an existing object model. In doing so, the new object inherits features from existing objects, making object-oriented programs easier to modify, more cost effective, and providing a quicker time to market. In summary, fast prototyping, easy migration to strongly typed objects, content reuse, separation of roles and responsibility for creating content, dedicated / specific toolset for the task at hand. [SR2iteRefresh](#) – design for success from the bottom up.

