

INTERLOOM™ WEB DEVELOPMENT FRAMEWORK

The InterLoom™ e-business solution is based on the real-world requirements faced by DataHouse customers. As more business applications become Web-centric, employees, customers and business partners put even greater demands upon the Web development framework of the enterprise.

Built with J2EE and XML, InterLoom™ brings the power of object-oriented programming to the Web development environment. Web developers can reap the OOP benefits of object inheritance, composition and polymorphism by using the InterLoom™ system.

The primary elements of the InterLoom Web Development Framework are templates and objects. The other InterLoom™ assets – beans, scripts and filters – enhance the functionality of objects and pages. Global resources are raw data elements that may be used by objects. The compiled logic, presentation and content are delivered to the Web in the form of pages.

APPLICATION INTERFACE

InterLoom™ Editor – Developers access the Web Development Framework through the same browser-based InterLoom™ Editor used to for the Content Management Tools. The various Web development assets are organized around a collection of managers. A pull down menu navigates the developer to the indicated manager.



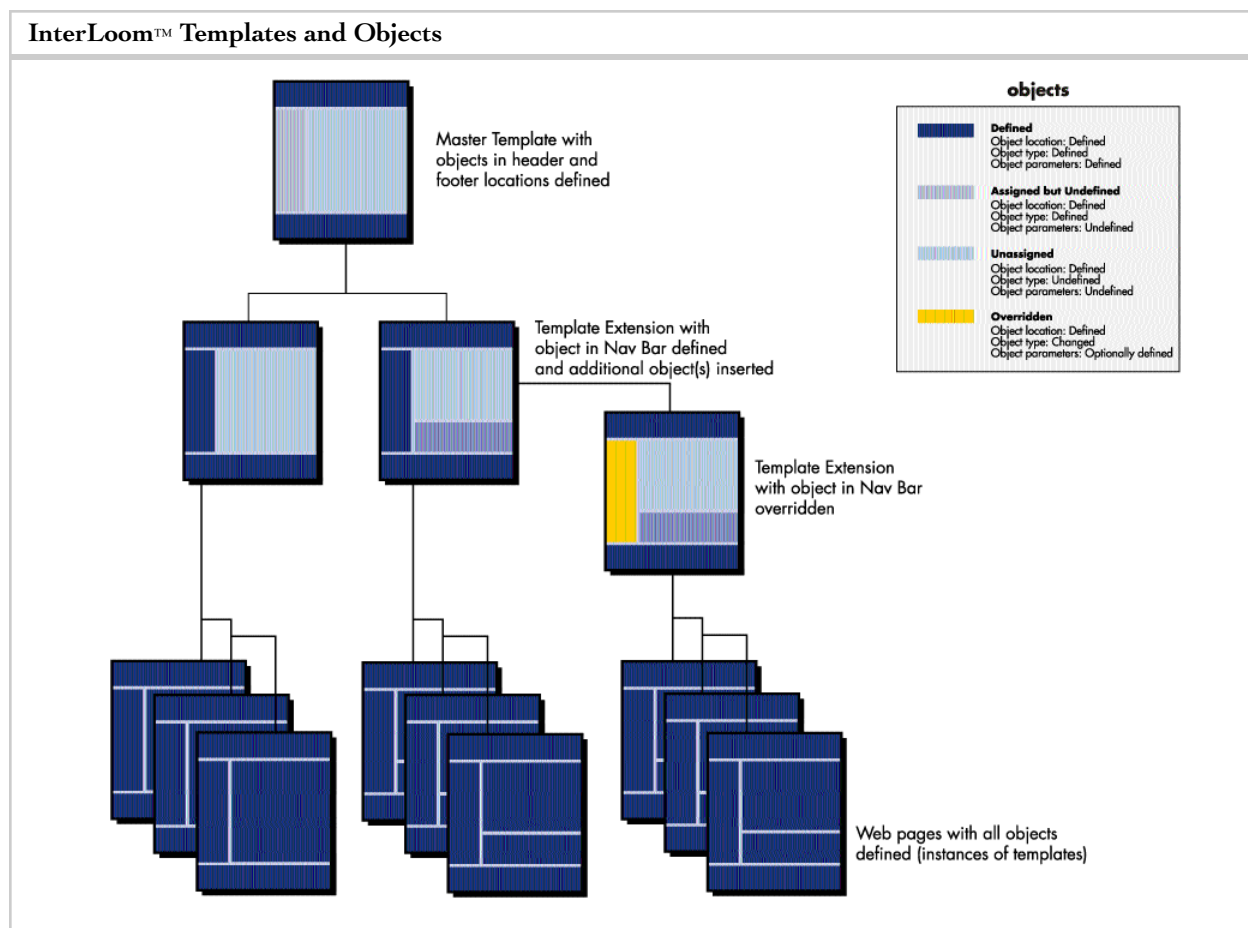
The browser-based InterLoom™ Editor greatly enhances team development by allowing developers spread across the enterprise to work on the same application with the benefits of workflow, check-in/check-out and user-security.

Developers realize gains in productivity through the intuitive WYSIWYG features of the InterLoom™ Editor. Contextual menus, toolbars and virtual file explorer features further enhance developer productivity.

INTERLOOM™ ASSETS

InterLoom™ Templates – Templates define the visual and programmatic structure of a Web application by assigning the objects and global elements that appear on a page and their location on the page. InterLoom’s hierarchical template inheritance feature creates a robust site architecture that is simpler to manage than the traditional file-driven model of Web applications. Developers can easily customize the look and feel of the application at a high level in the template hierarchy and allow it to cascade throughout the application. InterLoom™ supports overriding and granular customization at the lower levels of the hierarchy.

InterLoom™ Objects – Objects are a very powerful way of encapsulating and managing Web content and functionality. Instances of objects are created for various elements of the application and are assigned to tags that define where they are displayed and how they execute. Using objects accelerates development by enabling developers to assign objects to a template, extension or page instead of hand-coding every element. If the object needs to be changed (for example to reflect a change in the HTML standard), the object is changed only once and the changes will automatically be reflected everywhere the object was used. Objects control both the presentation of global resources and the management of the logic for features such as security, application integration and database connectivity. Core InterLoom provides a wide variety of objects for building the Web application. For complete listing of Core InterLoom™ objects, see Appendix A.



*In this example, the **master template** has defined the header and footer and created placeholders for the navigation bar and mainbody. To update the header or footer for the entire site, only the master template needs to be modified. At the next level, the **template extensions** inherit the header and footer from the master template. One template extension defines the object type for the navigation bar while the other extension **overrides** the object type for the navigation bar and **inserts** an additional object into the mainbody. This is ideal for creating consistent sub-sections of a Web site and offers flexibility within the template architecture. Finally, at the **Web page level** all the objects on the page have been defined (content has been created in the objects).*

The InterLoom™ object framework accelerates Web development by providing the following:

- **Inheritance** – Inheriting the behavior of an existing object can create a new object.
- **Composition** – A new object can be created that consists of a collection of existing objects.
- **Central Store** – By having a central store for objects, they can be included on any template making it easier to reuse the object throughout the site.
- **Reuse** – Objects can be reused throughout a site by having a central store of objects
- **Open Interface** – An open interface is provided so that new objects can be created.
- **Pre-built Objects** – Included with Core InterLoom™ are over twenty pre-built objects such as Article, Web poll, Search Input/Output and Database Report/Edit/Populate.
- **Custom Objects** – Custom objects can be developed in the InterLoom™ Application Framework to perform unique functions. Examples of custom objects include: zip code saver, automobile configurator, and user authenticator.

InterLoom™ Global Resources – Whereas an object encapsulates the data and functionality of an element in a Web application, global resources are the actual elements themselves.

Examples include jpeg, gif, Flash, video, MS Office files and blocks of HTML or Javascript code. InterLoom™ supports a wide variety of file types. Global resources are available for reuse throughout the Web site. Because global resources are stored in a central library, changes to the resource can be made in a single location. In addition, performance is improved by the use of a shared global resource that has already been cached by the system. Having resources available globally speeds development because resources are developed and deployed in only one area greatly reducing maintenance time.

InterLoom™ Beans, Scripts and Filters – These elements encapsulate server-side programming logic within the object-oriented InterLoom™ framework. These server-side elements add value to the e-business system by increasing levels of service, security and customization. They allow developers to apply data processing elements, interact with databases, error handling, and validation and provide custom interfaces to their applications.

InterLoom™ offers pre-coded functionality in the form of beans that can be applied to other objects such as forms and database connections. Beans reflect the modular programming abilities of J2EE. For more customized programming, developers can create server-side scripts using Java, JavaScript, Python or TCL. These server-side scripts can then be applied in an object-oriented fashion. Filters permit pre- and post-processing on a Web page. These filters can be custom scripted to implement features such as security, page counters or page re-direction.

- **InterLoom™ Beans** – InterLoom™ allows developers to utilize a variety of customized JavaBeans in their Web applications
- **Conditional Beans** – Browser Detector, Document Identity and Folder membership.
- **Populate Beans** – Database Populator, Multipopulator and Session Populator
- **Validation Bean**
- **Process Beans** – Database Field Saver, Forwarder, Login, Mailer, Multiprocessor, Searcher and Session Field Saver

- **InterLoom™ Scripts** – InterLoom™ allows developers to write custom scripts in Java, JavaScript, Python or TCL. The scripts are kept in a script library and can be called with an import directive. Objects that process HTML can be set to also interpret a specified programming language so that objects and functions can be called within the HTML code, along with an imported script.
- **InterLoom™ Filters** – Filters permit pre- and post-processing to Web pages.
- **Script Filter**– Allows for custom scripting of the filter
- **Session Check** – Provides security to selected Web pages
- **Virtual Host** – Forwards a user to another page based on a virtual host
- **Multifilter** – Allows several filters to be applied to a page

InterLoom™ Pages – InterLoom™ Server delivers Web pages by creating instances of the templates that the pages are derived from, and by including the objects assigned to them. InterLoom's production environment performs page-caching to decrease page-load time. Since pages are protected from programmatic change by InterLoom's template hierarchy, content creation and maintenance becomes trivial.

Web Development	
Feature	Benefit
Object-Oriented Model	The object model allows for effective re-use of designs, content and code.
Structural Views	Displays a virtual file directory for a clear view of the site and its components. Simplifies site administration and clarifies publishing areas.
Template Inheritance	This means that changes can be made at high levels in the inheritance tree and update all the levels below. This results in consistent designs with changes affecting as few templates as possible.
Template Swapping	Makes it is possible to switch the template that a document is based on, even during run-time. Provides greater flexibility in changing site structure.
Object Inheritance	Objects support inheritance, which means new objects can be created that extend the behavior of existing objects. It also allows instantiations of objects to inherit changes made to the original object.
Object Composition	A new object can be created that consists of a collection of existing objects. This simplifies the creation of more complex elements in Web applications.
Object Replacement	Object replacement on documents (object swapping).
Object Reuse	An object repository provides a central store for objects so they can easily be catalogued for use throughout the site.
Customizable Objects	Customizable objects allow site designers to use a simple form-based interface to configure objects.
Encapsulated Logic	Beans, scripts and filters allow developers to implement server-side programming logic in a modular, object-oriented fashion.
XML Support	Leverage XSL style sheets to create flexible XML documents that separate presentation from data and functionality.
Multi-format Output	Publish content in a variety of formats to a variety of devices, e.g. publish the same content to the Web (HTML), printer (PDF/printer-friendly), phone (WAP), PDA (WML).
Messaging	Framework to add messaging to Web sites. For example, auto-responders send automated responses when users submit forms.
Help System	Built-in help system quickly explains relevant features.