

ColdBox Platform Roadmap

Date: February 12, 2014



Introduction

The ColdBox Platform started as a scalable MVC framework out of the necessity for performance and stability. What set it apart from other frameworks was the productivity and simplicity of conventions-over-configuration and the broad range of tools that shipped with it. Over the years, these tools have evolved and matured into their own libraries that can be used outside of ColdBox MVC. ColdBox has also led the way in extensibility by providing core hooks via interceptors, sharable utilities via plugins, and truly modular applications with modules. ColdBox has also strived to create an open source ecosystem by providing ForgeBox as a centralized depot for developers to share code.

The development world is currently characterized by platforms that encourage simplicity, productivity, and a well-defined system for packaging code in a collaborative way. In turn, the ColdBox Platform aims to enable CFML developers in those areas as well as provide first-class tooling that encourages automation, testing, and distributed system architectures.

Development Areas

ColdBox 3.0 (March 2011) was a major release of the ColdBox platform that involved sweeping improvements to the internal architecture, identities for the LogBox, WireBox, and CacheBox libraries, and performance. ColdBox 4.0 will also be a major release for the platform with focuses on the following main areas of development.

Internal processes

ColdBox has always maintained a comprehensive set of unit tests to verify stability of new development and prevent regressions. The team has also used Ant build scripts to automate the build processes. With the advent of TestBox for xUnit and BDD style testing, we will switch our tests to run on TestBox. A proper build pipeline will be set up in Jenkins on our public staging server with post commit hooks to fully automate our unit tests and builds. This will provide the consistency necessary to facilitate our core revamp and will provide immediate bleeding edge builds of all our libraries.

ColdBox Core Revamp

Over the years, the ColdBox core has received many new features and additions which have increased the size and complexity of the platform. Last year we released ColdBox LITE for people who only wanted to use the MVC portions of the platform without the rest of the tooling. Going forward, all the internal features will be modularized and removed from the core such that ColdBox LITE will BECOME ColdBox and everything else will be an add on.

We feel this is necessary to keep the MVC core lightning fast and simple. In the same manner that jQuery has a core framework surrounded by an ecosystem of plugins, ColdBox will allow developers to pick and choose the features and libraries that are necessary for their application. We feel that modules are the future of package-driven CFML development. ForgeBox will be the central hub for all packages. Package dependency and even command line (CLI) interfaces will be introduced to facilitate developers in installing, configuring, and testing their applications in a manner similar to Ruby on Rails or Node.

New Features

ColdBox and its child Box libraries will receive updated look and feel for error templates and debugging screens, as well as changes aimed at making ColdBox apps "secure by default". We will focus on improving how routing is handled, and polishing how modules integrate with parent applications. Also planned are additional asynchronous functionalities, and more use of closures to create more expressive code.

Deprecations

In order to push the platform forward and keep it relevant, Team ColdBox is willing to remove functionality when it no longer serves a purpose. ColdBox 4.0 will see a number of deprecations, and eventually removals in our followup release of ColdBox 4.1. Many system plugins and interceptors are no longer needed and exist only for backwards compatibility from the 3.0 release 3 years ago. We are also dropping support for ColdFusion 8.

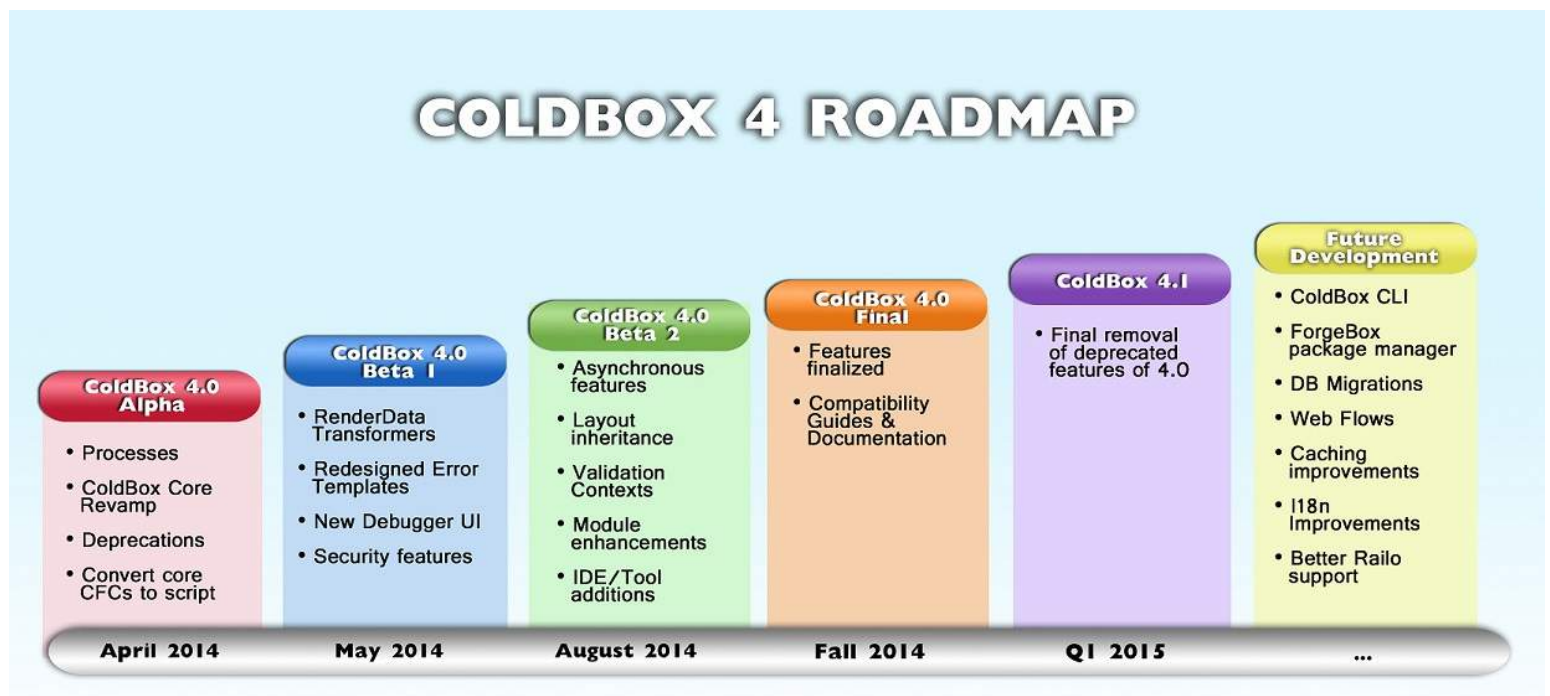
The goal is for 4.0 to be as backwards compatible as possible to give developers time to update code around deprecated features prior to their final removal in 4.1. We realize that this will create a bit more work for some applications to be upgraded, but we feel it is important to forge ahead and not let legacy functionality bloat the platform.

The concept of plugins will be entirely removed from ColdBox in favor of simple model objects or full-blown modules. Plugins are really no more than a CFC but when they were created neither WireBox or Modules existed. Now we believe the future of sharable "packages" of CFML code is modules. For simple utilities that truly need no more than a single CFC to exist, they will be treated as regular models; stored in a folder, mapped in WireBox and requested/wired when needed.

Development Tools

One of the areas we think CFML lags behind other development communities is in IDE integrations and tooling. We plan to build a Node-based CLI for packaging and deploying apps, installing packages from ForgeBox, and automating tests. We will also be releasing enhancements to our ColdBox Platform Utilities, as well as new syntax libraries and helpers for Sublime Text. Another focus will be on integrating TestBox more tightly into the development workflow.

Roadmap Graphic



Target Release Dates

ColdBox 4.0 Alpha April 2014

ColdBox 4.0 Beta 1 May 2014 (For **Into The Box**)

ColdBox 4.0 Beta 2 August 2014

ColdBox 4.0 Final Fall 2014

ColdBox 4.1 Q1 2015

Milestones

4.0 Alpha

Processes

- Automate ColdBox builds on commit using Jenkins Staging server
- Run unit tests and send report if failed
- Create bleeding edge build

ColdBox Core Revamp

- Create core modules and models convention
- Deprecate
 - Plugin methods and Wirebox DSLs
 - Reactor/Transfer support
 - BeanFactory plugin
 - Logger plugin
 - ObjectCacheManager (OCM) plugin
- Convert to modules/models
 - Core Debugger
 - AOP
 - Testing
 - Security Interceptor
 - All core plugins
- Convert core CFCs to script

4.0 Beta 1

ColdBox New Features

- Module auto namespace registration for model mappings
- RenderData Transformers
- Redesign default ColdBox Error template.
- Create "secure" and "public" error templates with secure as the default
- New UI for Debugger
- Allow .zip and .box extension on modules
- Form Anti Forgery tokens
- Named routes
- Add onInvalid HTTPMethod for REST
- Per-action exception handlers

4.0 Beta 2

ColdBox New Features

- Async Workers
- Async Events
- Validation Contexts to apply different rules in different situations
- Layout inheritance
- Security AOP & module enhancements
- Modules of modules

IDE/Tools

- CLI & ForgeBox integration
- ColdBox Platform Utilities enhancements
- Sublime integration enhancements.
- TestBox
 - CLI Runners
 - Gherkins
 - Node Runner
 - Grunt Runner

4.0 Final

- Features finalized
- Compatibility Guides created
- Documentation updated
- Sample Apps updated

ColdBox 4.1

- Remove functionality deprecated in 4.0

Future Development

- Coldbox CLI enhancements
- ForgeBox package manager
- Module dependencies
- Rails-like DB migrations
- Cachebox thread runners
- CFThread advice
- Resource bundle in DB + interface
- Event caching specify cache provider
- Event caching specify allowed RC
- View caching flag in config
- Support Railo strict scope
- Support Railo full null support
- Web flows

The [ColdBox Platform](#) is a copyright-trademark software by [Ortus Solutions, Corp](#)
Ortus provides [professional support and services](#) for all its open source products.
Ortus does encourage sponsorship of features that benefit your company.

All features and dates in this document are subject to additions, removals, and changes. This document is intended for informational purposes only, and should not be interpreted as a commitment on the part of Ortus Solutions, Corp. Ortus Solutions, Corp makes no warranties, express or implied, in this document.