

Release Plan (2.1 legacy document)

Table of contents

1 Comments.....	4
-----------------	---

Table of contents

1 Timeframe.....	3
2 Roadmap.....	3
3 The Future.....	3

Warning:

This document was copied as is from the Cocoon 2.1 documentation, but has not yet been fully reviewed or moved to its new home.

1. Timeframe

This is the current time frame for the next releases:

- Not scheduled yet: 2.1.7 (maintenance release)
- Not scheduled yet: 2.2M1 (Milestone 1)

2. Roadmap

In general, the next Cocoon version 2.2 can be seen as the "consolidation version". In the past a lot of new features, blocks and components were added which resulted in the fact that in some places concurring solutions were developed that could be combined a little bit.

One good example is form handling where we now are concentrating on CForms. We will try this in other areas as well. Of course this does not mean that we won't add new features to 2.2, but we try to consolidate what we currently have as well.

The follow is a list of items that are planned for the next minor release of Cocoon (2.2). This is a dynamic list that might change over time :)

We have come to the conclusion that real blocks might take a little bit longer than we have expected. So we agreed on not affecting the current development of Cocoon by the real blocks development. Which means we have to develop the blocks system separately from Cocoon until the block system is stable enough to be "merged" into Cocoon. This allows us to release new versions of Cocoon (like 2.2) without waiting for blocks to be working/finished.

- Official Versioning Guide (will be published soon).
- Virtual Sitemap Components.
- First finished version of CForms.
- Deprecate (not remove!) XSP (and provide a viable alternative).
- Cleaning up the caching/store mess.
- Deprecate blocks that haven't been maintained in a long while or don't serve any evident purpose. Web3, Apples, Python, Asciiart might be some candidates. We will decide this on a case-by-case basis.
- Differentiate between blocks that provide a service to other blocks and blocks that contain just samples or small applications built upon cocoon (petstore, tour, linotype). Maybe "samples-only" blocks should be a separate download. Perhaps remove deprecated blocks etc.
- Review the implications and the implementation of pooling.
- Reduce unneeded dependencies on Avalon, where possible.
- Review the logging framework. Log4j is the de-facto standard and we have blocks that complain if Log4j is not properly configured, so let's accept it and stop reinventing the wheel.
- Write more tests (you knew this one was coming ;-)).

The [updating to 2.2](#) document contains more information on how to prepare your application for the next minor release.

3. The Future

We already have plans for further versions of Cocoon that might be long in the future...

- Design and implement blocks.
- Integrate Kernel into the Cocoon core.

1. Comments

add your comments