

# Project description MMBase Component Framework

Author: André van Toly  
[andre@toly.nl](mailto:andre@toly.nl)  
Version: 0.9  
Date: 5 September 2006

<b>Version</b>	
<b>Title</b>	Project description MMBase Component Framework
<b>Status</b>	First draft
<b>Date</b>	05/09/2006
<b>Authors</b>	André van Toly

<b>Version</b>	<b>Date</b>	<b>Author</b>	<b>Comments</b>
0.2	02/07/2006	André van Toly	Initial version
0.3	28/08/2006	André van Toly	
0.9	05/09/2006	André van Toly	First draft

<b>Date</b>	<b>Still to-do</b>	<b>Version</b>
02/07/2006	Further specification of resource planning	0.2
05/09/2006	Review of several sections and resource planning	0.9

# Table Of Contents

<b>PREFACE.....</b>	<b>4</b>
<b>INTRODUCTION.....</b>	<b>5</b>
<b>GOAL.....</b>	<b>6</b>
Goals.....	6
Scope.....	6
Responsibilities.....	6
Risks and mediation.....	7
<b>DESCRIPTION.....</b>	<b>8</b>
Framework.....	8
Components.....	8
The development process.....	9
<b>DELIVERABLES.....</b>	<b>10</b>
Requirements or technical document.....	10
General document about the components and the framework.....	10
Interfaces to implement your framework.....	10
Reference framework.....	10
An example component.....	10
Test-cases and testing.....	10
Documentation.....	10
<b>PLANNING.....</b>	<b>11</b>
<b>APPENDIXES.....</b>	<b>12</b>
Reference, links, etc.....	12
People.....	12

# 1 Preface

This is the Project Document describing the MMBase Component Framework. This document does not follow exactly the structure laid down in a standard MMBase Project Document<sup>1</sup>, but rather follows a more traditional form of writing a project outline since we like to include sections as planning, resources etc. The sections 'Introduction' and 'Goal' can be found in this document, the missing sections 'Installation', 'Configuration', 'How to..', 'Functional architecture overview' and 'Technical architecture overview' are covered in the technical design documents of this project.

Since the project started recently this document is a work in progress.

---

<sup>1</sup> See the MMDocs under 'Writing documentation'.

## 2 Introduction

The content-management system 'MMBase' has now been Open Source for more than 6 years, and many sites have been built since. However, every CMS implementation on top of MMBase has been a process of reinventing the wheel: this is mainly because MMBase currently is more of a CMS toolkit than a complete CMS.

Many implementations now suffer from the problem that it is virtually impossible to share components that are built with MMBase, because of architectural choices that differ between systems.

The MMBase Component Framework project was started to address these issues. This document refers a lot to the technical discussion document "An MMBase Component Framework"<sup>2</sup>, which was the result of discussions held between both end-users and developers. It proposes a framework on top of MMBase that will allow components to be build in a generic way that allows easy incorporation into another framework. This project will try to realize that.

---

<sup>2</sup> "An MMBase Component Framework" (22 May 2006 by Johannes Verelst) contains the technical outlines or designs of this project.

## 3 Goal

The goal of this project is in short to develop software that makes it possible to 'run' components that communicate with MMBase and the 'application' – f.e. a CMS – that uses MMBase as its repository regardless of the 'application'.

### 3.1 Goals

This goal can be divided into several other main goals:

1. To write a technical description of the framework and the components it will harbor. Describing the way the generic framework will function, its components, considering the use of portlets<sup>3</sup> within components or vice-versa.
2. To write a formal component requirement document that contains formal rules on components that will allow third parties to develop them.
3. The development of a framework, or a set of interfaces that enable the development of a framework, that can harbor components regardless of its MMBase 'environment'.
4. The development of one or more components and portlets for testing, reference and demonstration purposes including editors to manage them.
5. The development of a reference framework which is needed to test the idea of the framework and the components themselves. It will also serve to test future components in order to be able to label them as 'MMBase Component Framework Compatible'.
6. To write documentation about everything mentioned before.

### 3.2 Scope

Not within the scope of this project is to develop a framework for one of the current CMS's or applications that were built on MMBase. The framework's description will be generic and when it focuses on a particular framework it will be the reference framework.

The development and imbedding of components in existing CMS's will be the task of the developers of the CMS's themselves.

The software will be developed upon MMBase version 1.8 and shall not be backwards compatible with earlier MMBase versions. The concluding application will be released with MMBase 1.9.

### 3.3 Responsibilities

Although there have been considerable strong wishes from several parts of the MMBase community – e.g. the MMBase Foundation, its partners and end users – to design and develop component architecture for MMBase, it is still the sole responsibility of the MMBase developers to develop such an

---

<sup>3</sup> Portlets are pieces of functionality that can be implemented in a web page. They are formally described in the JSR-168 specification. See paragraph 4.2.

architecture or framework. The MMBase Foundation has a supporting role in the process.

The other parties: the MMBase foundation, its partners and the end users, can be defined as stake-holders in this project.

### 3.4 Risks and mediation

The following risks can be identified.

Risk	Mediation
There is not enough time or manpower available to conclude the development of the framework and its components within a reasonable span of time.	Get more people involved.
Some of the developers will be able to develop on this project in their boss' time. Others, for example freelancers, can only work in their own spare time on this project.	
The project results become biased in the direction of the companies at which most of the development work is done.	The 'independent' developers need to have a critical view on the development process.
The endresult at the conclusion of this project is not what everybody wished for.	Keep 'others' outside the project involved.
etc.	

## 4 Description

In this section the framework, components and project itself will be described. Since the framework will form the embedding of the components, it is the topic of the first paragraph.

### 4.1 Framework

The MMBase Component Framework will not be the framework that runs on your MMBase instance that allows you to use components. That framework you will have to build yourself. Or rather you will have to implement the MMBase Component Framework and build a framework of your own that suits the way you use MMBase.

The idea is not to create a framework with a specific interface, but to only define interfaces and leave the implementation open. This requires significantly more effort in building a good set of interfaces, but as a result it is implementation-free.

An example implementation of the framework will be build upon a default MMBase distribution. This will also be used for testing the first components and will be the reference implementation.

This way existing MMBase sites can all migrate to this new framework and will hopefully minimize the amount of work required to incorporate generic components.

The result is that any website can integrate components, as long as a site-specific implementation of the framework interfaces is created. This means that for example the EO needs to write an implementation for their PATMOS MMBase framework, Kennisnet will need to write one for their Copacabana CMS.

During some technical sessions, consensus was reached about which interfaces should be defined for an “MMBase compliant CMS framework”.

The next few subsections will go deeper into these interfaces, but without too many technical details. This is on purpose, because we wish to define the exact details of these interfaces in a community-process. Our estimate is that this implementation will not be much larger than one or two pages of code.

### 4.2 Components

In short a component is an application using MMBase as its repository and written for MMBase. In the technical document, a component is defined as a set of resources, which together deliver a specific functionality.

An important requirement of the framework is that it will facilitate components to be aware of each other, to be able to communicate. And components need not to hinder each other and be able to act independently.

Another requirement, that will in most part be filled in by the developers who are also working at the CMS container, is that components can be used as portlets like the portlets described in the JSR-168 specification<sup>4</sup>.

All of these requirements will be specified in a special document targeted to allow third parties to write compliant components.

### 4.3 The development process

During the first project meeting held at the EO June 8th, it was concluded that development should be an iterative process. The development of the framework is considered to be rather complex. It can not be done without continuous testing and shaping. The following broad steps can be identified:

1. Guidelines and requirements, like configuration files, configuration directories, other directory structures, package names etc. will be described.
2. Some proposal framework interfaces will be distracted from existing frameworks like Didactor and the CMS-container.
3. A (skeleton) component will be made to test the first interfaces.
4. During coding all decisions made will be written down in documents stating the requirements of the framework and components.
5. During next iterations these steps will be broadly repeated.

To-do's:

- Create a template document/log for both the framework requirements and the component requirements during development?
- First description of configuration etc. (ad. 1).

---

<sup>4</sup> "Portlets are web components – like servlets – specifically designed to be aggregated in the context of a composite page. Usually, many portlets are invoked to in the single request of a portal page. Each portlet produces a fragment of markup that is combined with the markup of other portlets, all within the portal page markup." (from the Portlet Specification, JSR 168)":  
<http://www.jcp.org/en/jsr/detail?id=168>

## **5 Deliverables**

The deliverables, the 'products' of this project, can be deduced from the main goals of this project. These deliverables can be put on a timeline and can get resources attributed to them in the following chapter.

### **5.1 Requirements or technical document**

One of the most important outcomes of this project will be a document describing what components are and how the framework will work. Thus allowing third parties to write compliant components and implement a framework for them into their own MMBase application or CMS. These requirements probably will need to be very strict or they will not be able to guarantee that components can be installed across different MMBase installations.

This document will be technical of nature.

### **5.2 General document about the components and the framework**

Besides a technical document describing the framework and the interfaces, a more general document will be needed introducing the framework. This document will be more like an introduction or tutorial.

### **5.3 Interfaces to implement your framework**

The "MMBase Component Framework" will consist of a set of interfaces, the package `org.mmbase.framework`, that you will need to implement to construct your own framework that enables you to run components on your MMBase instance.

### **5.4 Reference framework**

The reference framework, being an example implementation of the aforementioned interfaces shall be developed. This framework will also be used during to test components.

### **5.5 An example component**

One or more example components will be developed, including JAR's, templates etc.

### **5.6 Test-cases and testing**

To test the functionality of the reference framework and its components test-cases will be developed.

### **5.7 Documentation**

Besides the documentation mentioned in the previous points the 'normal' MMBase documentation needs to be written.

## 6 Planning

The development process during the project will be iterative but the following broad steps can be determined.

#	Tasks	Time needed	
1	<b>Planning</b> - working on, continuous adjusting this document	1 - 2 weeks	
2	<b>Technical document (5.1)</b> - requirements like configuration etc. - technical descriptions	2 weeks	
3	<b>General document (5.2)</b> all of this in a more general introduction	2 - 3 weeks	
4	<b>Start of development</b> - first configuration files and directories - choosing example component(s) - creation of packages - first iterations of building framework interfaces	2 weeks	
5	<b>Developing the framework (5.3 &amp; 5.4)</b> - development of the interfaces - development of the reference framework	4 weeks	
6	<b>Example components (5.5)</b> - developing components - testing the framework using them	2 - 4 weeks	
7	<b>Test-cases and testing (5.6)</b>	1 week	
8	<b>Documentation (5.7)</b>	2 weeks	
9	<b>Finishing up</b> - Does the framework meet its requirements? - Is their sufficient documentation? - Can people develop their own framework? - Can third parties develop components?	2 weeks	+
<b>Total</b>			
		<b>18 - 22 weeks</b>	

Some steps mentioned above are already partly taken. Some tasks - e.g. 7, 8 and 9 - can be performed by non-MMBase developers.

## 7 Appendixes

### 7.1 Reference, links, etc.

Projectpage at MMBase.org <http://www.mmbase.org/mmcf>

Portlet Specification, JSR 168 <http://www.jcp.org/en/jsr/detail?id=168>

### 7.2 People

The following people contribute to this project:

Johannes Verelst  
Nico Klasens  
Daniel Ockeloen  
Michiel Meeuwissen  
Ernst Bunders  
Pierre van Rooden  
Henk Hangyi  
André van Toly