

# DockingFrames 1.0.3 - Transition

Benjamin Sigg

March 8, 2008

## Contents

|          |                                  |          |
|----------|----------------------------------|----------|
| <b>1</b> | <b>Version 1.0.3</b>             | <b>2</b> |
| 1.1      | Incompatibilities                | 2        |
| 1.1.1    | DefaultKeyboardController        | 2        |
| 1.1.2    | DefaultDockable/DefaultCDockable | 2        |
| 1.1.3    | CDockableListener                | 2        |
| 1.1.4    | FlapDockStation                  | 3        |
| 1.1.5    | XML                              | 3        |
| 1.1.6    | DockTheme                        | 3        |
| 1.1.7    | DockFactory                      | 3        |
| 1.2      | Features                         | 3        |
| 1.2.1    | SplitDockStation                 | 3        |
| 1.2.2    | SplitLayoutManager               | 4        |
| 1.2.3    | CDockable resize lock            | 4        |
| 1.2.4    | FlapLayoutManager                | 4        |
| 1.2.5    | ColorManager/ColorScheme         | 4        |
| 1.2.6    | ColorMap                         | 4        |
| 1.2.7    | LookAndFeel                      | 4        |
| 1.2.8    | CDockable resize request         | 4        |

## Abstract

This document describes the most important changes between versions, and how developers should change their application in order to use new features. This document does not make any distinction between the core-library and the common-project. Not all changes are listed up in this document, only those enhancements which might be interesting for the majority of developers.

# 1 Version 1.0.3

Version 1.0.3 emphasizes on background enhancements. The API remains unchanged for most parts.

## 1.1 Incompatibilities

These changes break with the API from 1.0.2, clients must be change their interfaces in order to work properly.

### 1.1.1 DefaultKeyboardController

**Short** The class `DefaultkeyBoardController` has been renamed to `DefaultKeyboardController`

**Reason** The new name looks better

**Clients** Replace any occurrence of `DefaultkeyBoardController` to `DefaultKeyboardController`

### 1.1.2 DefaultDockable/DefaultCDockable

**Short** `DefaultDockable` and `DefaultCDockable` now have `BorderLayout` set as default `LayoutManager`

**Reason** `BorderLayout` is the most often used `LayoutManager`.

**Clients** If another `LayoutManager` than `BorderLayout` is needed, set it up.

### 1.1.3 CDockableListener

**Short** `CDockableListener` divided into `CDockableStateListener` and `CDockablePropertyListener`

**Reason** `CDockableListener` was to big. Most clients either need information about the state, or about the properties of a `CDockable`. The case that both informations are needed is seldom.

**Clients** Need to decide which listener they implement. Note that `CDockableAdapter` implements both listeners, but not all methods get invoked when the adapter is registered only as one kind of listener.

#### 1.1.4 FlapDockStation

**Short** FlapDockStations layout is stored in a new format. The xml format will do the transition automatically, but the `DataInput/OutputStream` will not work properly.

**Reason** the old format did not carry enough information

**Clients** Store the layout in xml-format and load it again to do the transition.

#### 1.1.5 XML

**Short** `XElement` now extends `XContainer`, and no longer `XAttribute`. `XAttribute` extends `XContainer` as well.

**Reason** An element of a xml file is not an attribute, that is now reflected in the class structure

**Clients** May need to replace some occurrences of `XAttribute` by `XContainer`

#### 1.1.6 DockTheme

**Short** The common-project uses its own set of `DockThemes`. Each theme `XTheme` gets replaced by `CXTheme`

**Reason** The new themes make use of the new `ColorMap`

**Clients** Should use the new themes when possible. The old themes will work, but the user will see less features.

#### 1.1.7 DockFactory

**Short** `DockFactories` can now create any `Object` they want, and are no longer required to create `DockLayouts`. `DockLayout` has been converted into a class that wraps the `Object` that was created by a `DockFactory`

**Reason** All `DockLayouts` need to do the same things, hence clients would need to write the same code over and over again. Clients have now more freedom in how to implement `DockFactory`

**Clients** Should remove all occurrences of `implements DockLayout` and the methods `set/getFactoryId` that were defined in `DockLayout`

### 1.2 Features

This is the set of new features.

#### 1.2.1 SplitDockStation

**Short** The tree of elements of a `SplitDockStation` is now accessible from outside and can be modified directly

**Reason** It is more intuitive to work directly with the tree, some new algorithms work on the tree and are easier to implement that way.

### 1.2.2 SplitLayoutManager

**Short** New `SplitLayoutManager` calculates where to drop, and how to divide, elements of a `SplitDockStation`

**Reason** New features, like the locked size of `CDockable`, were only possible if the behavior of a `SplitDockStation` can be changed on runtime.

### 1.2.3 CDockable resize lock

**Short** The size of a `CDockable` can be locked during resize of its parent. See `setResizeLocked`, a method of `AbstractCDockable`.

**Reason** This was a request from a user

### 1.2.4 FlapLayoutManager

**Short** `FlapDockStation` now uses `FlapLayoutManager` to arrange its children

**Reason** Exchangeable behavior was a requirement for new features in the common-project.

### 1.2.5 ColorManager/ColorScheme

**Short** Many graphical elements now use `ColorManager` and `ColorSchemes`

**Reason** Colors can now be exchanged by clients. The control goes deep, even the color of a single element can be exchanged without affecting other elements of the same kind.

### 1.2.6 ColorMap

**Short** `CDockable` uses a `ColorMap` to define special colors for tabs and titles that are related to the `CDockable`

**Reason** This was a request from a user

### 1.2.7 LookAndFeel

**Short** Changes of `LookAndFeel` noted by `DockController` and forwarded to all `UIListeners`.

**Reason** Because the `ColorManager` would not be informed of the new `LookAndFeel` otherwise

### 1.2.8 CDockable resize request

**Short** `CDockables` can now request a size they would like to have, and in most environments they will get this size. See the method `setResizeRequest` of `AbstractCDockable`.

**Reason** This was a request from a user