Getting Started with Knowledge Management and Collaboration

The Knowledge Management and Collaboration (KMC) Platform consists of several layers, as illustrated in the following graphic:

**Repository Framework**

The lowest layer is the repository framework (RF) layer. It offers an extensible framework for various types of extensions.

- The most important type of extensions are the repository managers, which connect to the repositories in which the documents and other objects are stored and expose these documents in a unified fashion as resources. For example, the file system repository manager connects to a folder in the file system and exposes the files and folders therein as resources.
- Repository filters allow the manipulation of these resources while they are being passed through the framework. For example, a property filter might add additional metadata to a resource.
- In addition to the unified aspects of the repository managers, repository services can add additional aspects to the objects of a specific repository. For example, the application properties service stores additional data on resources in a database table.
- Global services supply additional functions for the resources exposed by the RF. For example, the relation service offers functions for the relations between two resources, such as "resource A is attached to resource B".
- Semantic objects offer a mechanism for "casting" a resource onto another object. For example, a resource that is also a team room can be "casted" onto a TeamRoom object.

The RF also provides a default component implementation (for example, the CM repository manager, the application property service, or the relation service). The RF is the foundation for the other layers and provides the runtime for the KMC components that use or extend the RF.
Between the repository framework and the Knowledge Management layer lies TREX. TREX implements Knowledge Management's powerful search and classification engine. However, since TREX does not provide officially released APIs for Java, it will not be described here in detail (see the Administration Manual for further information). Instead, the index management service of the Knowledge Management layer provides an API for using the search and classification functions of TREX.

Knowledge Management

The Knowledge Management (KM) layer uses the repository framework and adds further extensions that implement the functions for document management that are visible to the user (for example, repository managers such as the Web repository manager, repository services such as the subscription repository service, and global services such as the attachment service).

It also implements the flexible UI, which allows users to navigate through the document hierarchy. The flexible UI itself is also - like the RF - an extensible framework for FlexUI extensions, for example, collection renderers or resource renderers.

Finally, KM's index management service integrates the TREX search engine into Knowledge Management and offers extension points where additional search engines can be integrated.

Collaboration

The next layer, build upon the KM layer, is the Collaboration layer. It offers applications that allow users to collaborate within the portal, chat, share their documents, and so on. Like the KM layer, it brings its own extensions that implement the collaborative functions into the RF, as well as some extensions for the flexible UI (such as the team room renderer).

The extension types offered by Collaboration are:

- **Room extensions**, which allow the Collaboration team rooms to be extended by additional items.
- **Collaboration services**, which implement functions that can be launched by portal users, such as "send an e-mail" or "dial phone".
- **Groupware connectors** (or transports), which are connectors to groupware servers such as Lotus Notes or Microsoft Exchange.
- **Synchronous Collaboration Framework connectors**, which are connectors to meeting servers such as WebEX or NetMeeting.

KMC Application

An application on top of these layers can use any of the APIs offered by these layers, by usage (that is, implementing an application that retrieves all the documents in a given directory) or by extending them (for example, by implementing a repository manager or an flexible UI renderer).