Проект CM6-base

# Структура

Проект CM6-base (CompanyMedia 6 base edition) имеет следующую структуру каталогов, соответствующую структуре проектов и модулей maven:

* modules
  + classifier-module
  + sochi-integration
  + org-structure
  + base-documents
  + generic-documents
  + другие модули, использующиеся в проекте «Атлас»
* standalone
  + app
  + war
  + ear
* overall-build

Такая структура разработана с целью возможности двоякого использования модулей проекта. Первый вариант – отдельное enterprise-приложение (standalone ear), предназначенное как для тестирования модулей разработчиками CM6Base, так и для последующего развития в СЭД CompanyMedia 6. Второй вариант – в составе общего ear с Сочи-сервером, образуя гибридную СЭД CompanyMedia 5/6. В этом варианте используется только компонент modules. Компоненты для сборки общего приложения будут реализованы позднее.

# Модули

Полное описание состава модулей приведено в документе [Состав модулей CM6](http://git.inttrust.ru/git/?p=CM5.git;a=blob;f=%D0%A1%D0%AD%D0%94+CM6-base/%D0%A1%D0%BE%D1%81%D1%82%D0%B0%D0%B2+%D0%BC%D0%BE%D0%B4%D1%83%D0%BB%D0%B5%D0%B9+CM6.docx;h=6a2a4bab25da87c79c8f0e6390537b86d0fca1db;hb=HEAD).

## Classifier-module

Был разработан в составе проекта «Атлас», переиспользуется в CM6Base.

Maven artifactId: ru.intertrust.cm-base:classifier-module  
Корневой package: ru.intertrust.cm.classifier  
Зависимости: commons-module (в настоящее время; эту зависимость желательно удалить).

## Sochi-integration

Компоненты, обеспечивающие взаимодействие CM6 и Сочи-сервера.

Maven artifactId: ru.intertrust.cm-base:sochi-integration  
Корневой package: ru.intertrust.cm.integration.sochi  
Зависимости: ru.intertrust.cm-sochi:ssrv-sochi-fellow (конфигурация ДО Сочи); возможно, понадобятся другие зависимости (сервисы СО из Сочи, ПКД из CMJ)

## Org-structure

Модуль организационно-штатной структуры организации, а также внешних корреспондентов.

Maven artifactId: ru.intertrust.cm-base:org-structure  
Корневой package: ru.intertrust.cm.staff  
Зависимости: sochi-integration (для реализации на «бородах» из Сочи-сервера; для CM6 будет добавлена собственная реализация)

## Base-documents

Компоненты, реализующие общие функции документооборота.

Maven artifactId: ru.intertrust.cm-base:base-documents  
Корневой package: ru.intertrust.cm.documents.base  
Зависимости: org-structure

## Generic-documents

Модуль, содержащий реализацию простых (нерегистрируемых) документов.

Maven artifactId: ru.intertrust.cm-base:generic-documents  
Корневой package: ru.intertrust.cm.documents.generic  
Зависимости: base-documents, classifier-module

## App

Код и конфигурация, обеспечивающие работу модулей CM6Base в составе отдельного приложения.

Maven artifactId: ru.intertrust.cm-base:standalone-app  
Корневой package: ru.intertrust.cm.base.app  
Зависимости: ?

## War

Сборка web-приложения.

Maven artifactId: ru.intertrust.cm-base:standalone-war  
Корневой package: не должен содержать код  
Зависимости: generic-documents, реализация платформы AF5 (service-impl, dao-impl, gui-server-impl, gui-client-impl).

## Ear

Сборка enterprise-приложения.

Maven artifactId: ru.intertrust.cm-base:standalone-ear  
Корневой package: не должен содержать код  
Зависимости: standalone-war

## Overall-build

Служит как единая точка сборки всех модулей, входящих в CM6Base.

Maven artifactId: ru.intertrust.cm-base:overall-build  
Корневой package: не содержит код  
Зависимости: modules, standalone

# Инфраструктура

## Git

Код проекта располагается в основной (master) ветке репозитория git по адресу ssh://git@stash.inttrust.ru:7999/cm6b/cm6base.git (<https://stash.inttrust.ru:8443/projects/CM6B/repos/cm6base/browse>).

Коммиты в репозиторий должны осуществляться только от своего имени. Комментарий к каждому коммиту должен содержать идентификатор запроса JIRA, в рамках которого были внесены изменения, а также осмысленный текст, описывающий эти изменения. Коммиты без запроса JIRA допускаются только в крайних случаях при минимальных изменениях (исправление опечаток и т.п.) При обновлении локальной копии репозитория предпочтительно сливать изменения методом rebase, а не merge (см. [Стандарты кодирования](http://git.inttrust.ru/git/?p=CM5.git;a=blob;f=%D0%98%D0%BD%D1%84%D1%80%D0%B0+%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B0/%D0%A1%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82%D1%8B+%D0%BA%D0%BE%D0%B4%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F.docx;h=57dc1d980a2dd8a7ce658fb97e1490f7abf6a22c;hb=HEAD)). Служебные файлы среды разработки (Eclipse, IDEA и др.) в репозиторий помещаться не должны.

## JIRA

Отслеживание запросов по проекту ведётся в общем репозитории с проектом «Сочи»: <https://jira.inttrust.ru:8443/browse/CMFIVE/>, для модулей CM6Base заведён отдельный компонент – CM6-Base (<https://jira.inttrust.ru:8443/browse/CMFIVE/component/13221>). Любая разработка по проекту должна осуществляться только по запросу в JIRA. В запросе должны корректно проставляться версии, в которых реализован запрошенный функционал, а для ошибок – версии, где обнаружена ошибка (для запросов по новому функционалу в поле «Проявляется в версиях» лучше ставить псевдоверсию «Backlog»).

## Nexus

Собранные maven артефакты помещаются в один из двух Nexus-репозиториев Интертраст:

* <https://nexus.inttrust.ru/nexus/content/repositories/cm5-releases/> (intertrust.releases.cm-sochi) – релизные версии;
* <https://nexus.inttrust.ru/nexus/content/repositories/cm5-snapshots/> (intertrust.snapshots.cm-sochi) – snapshot-версии.

Выпуск версий осуществляется по команде руководителя проекта.

## Нумерация версий

Все модули проекта используют единый номер версии:

**6.0**-a*1*

Цифры, выделенные жирным (6.0) – постоянны. При выпуске версии меняется последнее число (после буквы a). Эта схема применяется на время первичной разработки; при приближении к состоянию готовности и передаче продукта заказчику (либо в ЦТС) будут выпускаться версии без литеры a (alpha) и, возможно, с большим количеством числовых компонент в номере.

Использование snapshot-версий компонентов, включая разработанные в компании «Интертраст», не допускается.