Reference code

Application design

Description of the design of the reference Java system developed as part of the project.

- Modules — Description of the application breakdown into separate modules
  - MessageXML module — XSD schemas defining the data structures of the Bitmagasin messages
  - Protocol module — Contains the general protocol functionality, like message bus connectors, message processing and data transmission code
  - Access client module — Contains functionality for reading from the BitMagasin. This includes Get File, Get Checksums and Get File IDs
  - Modifying client module — All functionality for writing to the Bitrepository is located here. This includes Put File, Delete File and Replace File.
  - Processing client module — Contains the client side functionality for Processing Single File and Mass Processing
  - Alarm client module — Contains the Alarm client functionality
  - Integrity client module — Contains the integrity client code
  - Audit client module — Contains the client side functionality for accessing audit logs.
  - Monitoring module — Contains the code for the central monitoring service, which is part of the Coordination Layer.
  - Integration module — The tests for examining the integrated system (or part of it).
  - Common module — Contains code shared by all modules. Examples are utils and configuration loading.

Building the Bitrepository

Here you can find information on how to use Maven based build environment to start development on the Bitrepository system.

- Building — Description in how to compile the Bitrepository source
- Common build problems
- Creating distributable — mvn package on the command line should do it.
- Handling licensing information — Maven takes care of this, see License Management
- Starting Coordination layer — Description on how to start a ActiveMQ and monitoring service for local testing.
- Testing — Description in how to test the Bitrepository system

Code guidelines

Description of the code design we use in the application development

- 3-layered components — All our components should be keep the data accessing (read and write), business logic and interface code separated
- Argument validation — Describes how and when to validate arguments
- Code style (Deprecated) — The general code style used in the java code based on the SBForge code style.
- Construction, referencing and composing in general — A short rattle of what to do and not to do when creating or getting references to objects
- Exceptions — We will lean on the Effective Java Exceptions http://www.oracle.com/technetwork/articles/entarch/effective-exceptions-092345.html guidelines.
- Logging — Here you can find a description of logging is implemented in the Bitrepository code
- Writing automatic tests — Guideline for writing good automatic test cases

Setup of development environment

Here you can the documentation need to setup a development environment