netarchive.dk

The Architecture of NetarchiveSuite

Søren V. Carlsen (svc@kb.dk)

NetarchiveSuite Java Modules (1)

- The software consists of the following modules:
  - Common (dk.netarkivet.common)
  - Harvester *(dk.netarkivet.harvester)*
  - Archive *(dk.netarkivet.archie)*
  - Viewerproxy *(dk.netarkivet.viewerproxy)*
  - Monitor *(dk.netarkivet.monitor)*
  - ”Deploy *(dk.netarkivet.deploy)*

NetarchiveSuite Java Modules (2)

- Common module:
  - Contain common functionality (like exceptions, JMS sending and receiving, ARC utilities) not specific to one module.
  - Declares most of the interfaces, that the plugins must implement (only exception DBSpecifics)
  - Applications: dk.netarkivet.common.webinterface.GUIApplication
- All other modules
  - depend on the common module,
  - but can otherwise be used independently

NetarchiveSuite Java Modules (3)

- Harvester module:
  - Contains software related to partitioning, scheduling, and running HarvestJobs.
  - Main apps:
    - dk.netarkivet.harvester.harvesting.HarvestControllerApplication (Accepts jobs, starts Heritrix, return results)
    - dk.netarkivet.harvester.sidekick.SideKick
    - dk.netarkivet.harvester.datamodel.HarvestTemplateApplication (Creates, downloads, updates, and deletes templates in database)
    - dk.netarkivet.harvester.webinterface.HarvestDefinitionApplication (GUIApplication + scheduler in one app)

NetarchiveSuite Java Modules (4)

- Archive module:
  - Contains software implementing an archive distributed among several locations.
  - Main Applications:
    - dk.netarkivet.archive.arcrepository.ARCRepository Application
    - dk.netarkivet.archive.bitarchive.BitarchiveApplicati on
    - dk.netarkivet.archive.bitarchive.BitarchiveMonitor Application
    - dk.netarkivet.archive.indexserver.IndexServerAppli cation

NetarchiveSuite Java Modules (5)

- Viewerproxy module:
  - Implements software to browse in archived material using a proxyserver.
  - Main Applications:
    - dk.netarkivet.viewerproxy.ViewerProxyApplication
- Monitor module:
  - Software that utilizes JMX to monitor the running NetarchiveSuite
  - Main Applications:
    - dk.netarkivet.monitor.tools.JMXProxy

NetarchiveSuite Java Modules (6)

- The modules and their associated webpages:
  - Harvester
    - webpages/HarvestDefinition (DefinitionsSitesection)
    - webpages/History
  - Archive (webpages/BitPreservation)
  - Viewerproxy (webpages/QA)
  - Monitor (webpages/Status)

Messages and datatransfer (1)

- Data is transmitted between applications either embedded inside messages, or using remoteFiles
- Messages is sent by applications, and received by application using a JMS message broker as middleman
- JMS messages can be sent to either a TOPIC Channel or a QUEUE Channel.
- In case of a TOPIC, all listeners will get the message. But if no one is listening, no one gets the message.

Messages and datatransfer (2)

- In case of a QUEUE channel, the message remains in the channel until a listener will accept the message, or the message is deleted when cleaning the JMSBroker during restart of NetarchiveSuite
- Some of the channels used:
  - THE_SCHED (one in all) – receives messages from the harvesters (QUEUE – channel)
  - ARC_REPOS (one in all) – QUEUE channel
    - Receives archive-requests (getRequests, UploadRequests) from clients
    - Forwards the requests to the archive.
  - ALL_BA (one for each location) – TOPIC channel
    - Used to multitask batchjobs to the BitarchiveServers

Messages and datatransfer (3)

- More channels used:
  - ANY_HIGHPRIORITY_HACO (one in all) – QUEUE channel
    - Scheduler sends selective harvest jobs to this channel
    - The HarvestControllers (with queuePriority set to HIGHPRIORITY) are listening to this channel, when they are idle
  - ANY_LOWPRIORITY_HACO (one in all) – QUEUE channel
    - Scheduler sends snapshot harvest jobs to this channel
    - The HarvestControllers (with queuePriority set to LOWPRIORITY) are listening to this channel, when they are idle

Messages and datatransfer (4)

- The Message channels is named with the JMS environment as prefix,
  - so multiple NetarchiveSuite instances can use the same JMS broker without interfering with each other
  - And to make it easy to empty any remaining messages lying in the channels during restart of the NetarchiveSuite

Implemented messages (1)

- All messages extend the class NetarkivetMessage
  - dk.netarkivet.common.distribute.NetarkivetMessage
- Harvester messages
  - Extend dk.netarkivet.harvester.distribute.HarvesterMessage
  - DoOneCrawlMessage (dk.netarkivet.harvester.harvesting.distribute.*)
  - CrawlStatusMessage (dk.netarkivet.harvester.harvesting.distribute.*)

Implemented messages (2)

- Archive messages
  - Extend dk.netarkivet.archive.distribute.ArchiveMessage (dk.netarkivet.harvester.harvesting.distribute.*)
  - Bitpreservation messages: (dk.netarkivet.archive.arcrepository.bitpreservation)
    - AdminDataMessage
    - RemoveAndGetFileMessage,
  - Batchmessages :
    - BatchMessage, BatchEndedMessage, BatchReplyMessage
  - Request messages:
    - GetFileMessage, GetMessage, IndexRequestMessage, StoreMessage, UploadMessage

Using JMSConnection (1)

- import dk.netarkivet.common.distribute.JMSConnection;
- JMSConnection con = JMSConnection getInstance();
- NetarkivetMessage nm = ...
- *con.send(nm)* - send the message (the message itself knows the correct destination queue or topic)
- *con.replyTo(nm)* - send message back to the sender of this message (or where ever we want the messages to go)

Using JMSConnection (2)

- ChannelID A = ..;
- con.resend(nm, A) - used to forward or resubmit a message to queue **A** when a listener accepts the message from a queue (which removes the message from queue), and then finds that he can't handle the message.
- con.setListener(ChannelID mq, MessageListener ml)
  - Start listening to channel mq using MessageListener ml
- con.removeListener(ChannelID mq, MessageListener ml)
  - Stop listening to channel mq using MessageListener ml
- All listeners must implement the interface javax.jms.MessageListener;
  - void onMessage(Message)

The layout of the package (1)

- The package contains
  - build.xml (ant file for building the code)
  - lib (3rdparty libraries required)
  - src (the Java source code)
  - tests (unittests, and integrity tests for the software, as well as additional 3rd party libraries required by the tests)
  - conf (default settings.xml, and deploy-script)
  - webpages (the GUI)
  - scripts/simple_harvest (Easy deploy of NetarchiveSuite)
  - scripts/sql - SQL create scripts for Derby and MySQL
  - docs (javadoc for NetarchiveSuite

The layout of the package (2)

- The package contains
  - harvestdefinitionbasedir/fullhddb.jar (default Derby database ready for use)
  - harvestdefinitionbasedir/order_templates (default harvest templates)

The build.xml and its targets

- Ant version: 1.6.2+
- We have targets that enables us to
  - run tests (unittest, fulltest)
  - generate a codecoverage report (clover.report)
  - generate the module jarfiles (jarfiles)
  - generate the webpages warfiles (warfiles)
  - generate the javadoc for the code (javadoc)
  - generate a NetarchiveSuite release package (releasezipball)
- Show the build.xml