netarchive.dk

Installation Discussion

Plugins in the NetarchiveSuite

What is plugins or pluggable settings?

- Some of our settings constitute plugins or pluggable settngs, where the default java class can be replaced with a class of your choice. The only condition is that it implements the same Java interface

- In most cases, the class setting itself is only part of the plugin configuration

netarchive.dk

RemoteFile plugin(1)

☐ Function:

Designates the method of filetransfer between applications in the NetarchiveSuite

☐ Settings (for FTPRemoteFile)

settings.common.remoteFile.class

settings.common.remoteFile.serverName

settings.common.remoteFile.serverPort

settings.common.remoteFile.userName

settings.common.remoteFile.userPassword

settings.common.remoteFile.retries

RemoteFile plugin(2)

- Settings (for HTTPRemoteFile):
  - settings.common.remoteFile.port

- Default class:
  - *dk.netarkivet.common.distribute.FTPRemoteFile*

- Alternative classes:
  - *dk.netarkivet.common.distribute.HTTPRemoteFile*

- Implements:
  - *dk.netarkivet.common.distribute.RemoteFile*

netarchive.dk

Sitesection plugin(1)

- Function: Defines a part of the webinterface. The class dk.netarkivet.common.webinterface.GUIWebserver reads all the sitesections in the settings.xml. Each sitesection contains

  - A *class*. Encapsulates metadat about the sitesection, e.g (metadata about the sitesection, e.g. the title of the sitesection, and the subpages of the sitesection.

  - A *webapplication* (the directory or war-file containing the webapplication for this sitesection)

  - A *deployPath* (The URL path for this sitesection)

- Settings (for sitesection)

  - setting.common.webinterface.sitesection.class

  - setting.common.webinterface.sitesection.webapplication

  - setting.common.webinterface.sitesection.deployPath

netarchive.dk

Sitesection plugin(2)

- Default classes:
  - 1. *dk.netarkivet.harvester.webinterface.DefinitionsSiteSection*
    Allows you to define and schedule harvests
  - 2. *dk.netarkivet.harvester.webinterface.HistorySiteSection*
    See the status of running and finished harvestjobs
  - 3. *dk.netarkivet.archive.webinterface.BitPreservationSiteSection*
    Allows you to check and correct data in our archive
  - 4. *dk.netarkivet.viewerproxy.webinterface.QASiteSection*
    Allows you to perform Quality Assurance
  - 5. *dk.netarkivet.monitor.webinterface.StatusSiteSection*
    Allows you to monitor the health of all machines and applications
- Interface: *dk.netarkivet.common.webinterface.SiteSection*

netarchive.dk

arcrepositoryClient plugin(1)

- ❑ Function: Upload and download data to and from archive

- ❑ Settings:
  - ◼ settings.common.arcrepositoryClient.class
  - ◼ settings.common.arcrepositoryClient.getTimeout
  - ◼ settings.common.arcrepositoryClient.storeTimeout
  - ◼ settings.common.arcrepositoryClient.storeRetries

netarchive.dk

arcrepositoryClient plugin(2)

- ☐ Default class:
    - ▪ *dk.netarkivet.archive.arcrepository.distribute.JMSArcRepositoryClient*
    - ▪ This class uploads to and downloads files from an archive using JMS Messages and RemoteFiles.

- ☐ Alternative class:
    - ▪ *dk.netarkivet.common.distribute.TrivialArcRepositoryClient*
    - ▪ A minimal implementation of ArcRepositoryClient that just has one local directory that it keeps its files in, no checking nothing

- ☐ Implements:
    - ▪ *dk.netarkivet.common.distribute.ArcRepositoryClient*

netarchive.dk

indexClient plugin (1)

- Function:

  Requests and receives harvest job indices.

- Settings:

  - *settings.common.indexClient.class*

  - *settings.common.indexClient.indexRequestTimeout (only used by default class)*

- Default class:

  - *dk.netarkivet.archive.indexserver.distribute.IndexRequestClient*

    This class requests and receives the indices from a IndexServerApplication. The communication is done using JMS, and RemoteFiles.

netarchive.dk

indexClient plugin (2)

☐ **Alternative class:**

- ▪ *dk.netarkivet.common.distribute.indexserver.TrivalJobIndex Cache*

  A trivial *JobIndexCache* implementation that just assumes somebody places the indices i a directory relative to the cache dir).

☐ **Implements:**

- ▪ *dk.netarkivet.distribute.indexserver.JobIndexCache*

JMS broker plugin (1)

❑ Function: JMS (Java Message Service) Client

❑ Settings:

- ❑ settings.common.jms.class (the plugin itself)
- ❑ settings.common.jms.broker (the host where the JMS broker resides)
- ❑ settings.common.jms.port (the port)
- ❑ settings.common.jms.environment (Environment - common prefix to all JMS channels created by the NetarchiveSuite)

❑ Default class:

- ▪ *SunMQ (*shorthand for dk.netarkivet.common.distribute.JMSConnectionSunMQ*)*

netarchive.dk

Notifications plugin (1)

❑ **Function:** Method of notification. In case of a serious problem, a notification is sent to a designated receiver.

❑ **Settings:**

    ❑ settings.common.notifications.class (the plugin)

    ❑ settings.common.notifications.receiver (the designated receiver)

    ❑ settings.common.notifications.sender (the sender)

    ❑ settings.common.mail.server (The mail-server to be used)

❑ **Default class:**

    ❑ *dk.netarkivet.common.utils.EmailNotifications*

    (Class using Email as notification method)

Notifications plugin (2)

- Alternative classes: None at this time

  (However for testing purposes there is also the class RememberNotifications)

- Interface:

  - *dk.netarkivet.common.utils.Notifications*

netarchive.dk

Database plugin (1)

❑ Function: Selects the database of your choice; and encapsulates differences between databases; e.g. the JDBC drivers

❑ Settings:

 ❑ settings.harvester.datamodel.database.url
 The full JDBC URL for connecting to the database

 ❑ settings.harvester.datamodel.database.specificsclass
 The plugin itself

 ❑ settings.harvester.datamodel.database.backupInitHour
 The earliest hour for initiating DB backup

netarchive.dk

Database plugin (2)

❑ ## Default class:

   ❑ *dk.netarkivet.harvester.datamodel.DerbyEmbeddedSpecifics*

   Derby as embedded database

❑ ## Alternative classes:

   ❑ *dk.netarkivet.harvester.datamodel.DerbyClientSpecifics*

   External Derby databaseserver

   ❑ *dk.netarkivet.harvester.datamodel.MySQLSpecifics*

   MySQL database (MySQL 5.0.x)

❑ ## Interface:

   ❑ *dk.netarkivet.harvester.datamodel.DBSpecifics*

netarchive.dk

More information about plugins

- Installation Manual, Appendix C
- Developer Manual (Section 6: Pluggable parts)