EFTS Agents Software Design Document

Doc.No. : EUM/OPS-MSG/TEN/04/0575
Issue : 1
Date : 5 November 2004

EUMETSAT
Am Kavalleriesand 31, D-64295 Darmstadt, Germany
Tel: +49 6151 807-7
Fax: +49 6151 807 555 Telex: 419 320 metsat d
http://www.eumetsat.de

Document Change Record

Issue / Revision	Date	DCN. No	Changed Pages / Paragraphs
1.0 Draft A	26/03/2003		Original Version - /EUM/OPS_MSG/TEN/03/0075 – EFTS PreProcessor Software Design Document.
1	5 November 2004		This document constitutes the final version for EFTS_AGENTS V2.20. It has been released under a new document number and template to comply with EUMETSAT document management procedures.

Table of Contents

1 Introduction 7
1.1 Reference Documents 7
1.2 Overview 7
1.3 Agents Overview 8
1.3.1 Poll Agent 8
1.3.2 PreProcessor Agent 9
1.3.3 Push Agent 10
1.3.4 Connection Checker Agent 10
1.4 Agent Execution Modes 10
1.5 Pre-Conditions 11
1.6 Constraints 11
2 Overview of Software 12
2.1 Agents 12
2.2 Operating Environment 12
3 Operational Configuration Isssues 13
3.1 Installation 13
3.1.1 General Information 13
3.1.2 Installation on a UNIX system 13
3.1.3 Installation on a Windows system 13
3.2 The Uninstall Process 14
3.3 Start Up and Shut Down of the Agents 15
3.3.1 Command line startup 15
3.3.2 Startup scripts on Windows Systems 15
3.4 Configuration 16
3.4.1 Housekeeping 17
3.4.2 Properties for the PreProcessor Agent 17
3.4.3 Configuration as a PushFtpAgent 19
3.4.4 Configuration as PollFtpAgent 20
3.5 Testing and Future Developments 21

1 INTRODUCTION

This document describes the design of the EFTS Agents for the Extended File Transfer Software (EFTS).

1.1 Reference Documents

RD [1] EFTS Interface Control Document, EUM/OPS/ICD/01/0712, 29th October 2003.
RD [2] GEMS Software Design Document, EUM/OPS/TEN/00/1454, 19th April 2004.

1.2 Overview

The EFTS Agents is a collection of processes that do local or remote file transfer, copying, and manipulation. It can be used to:

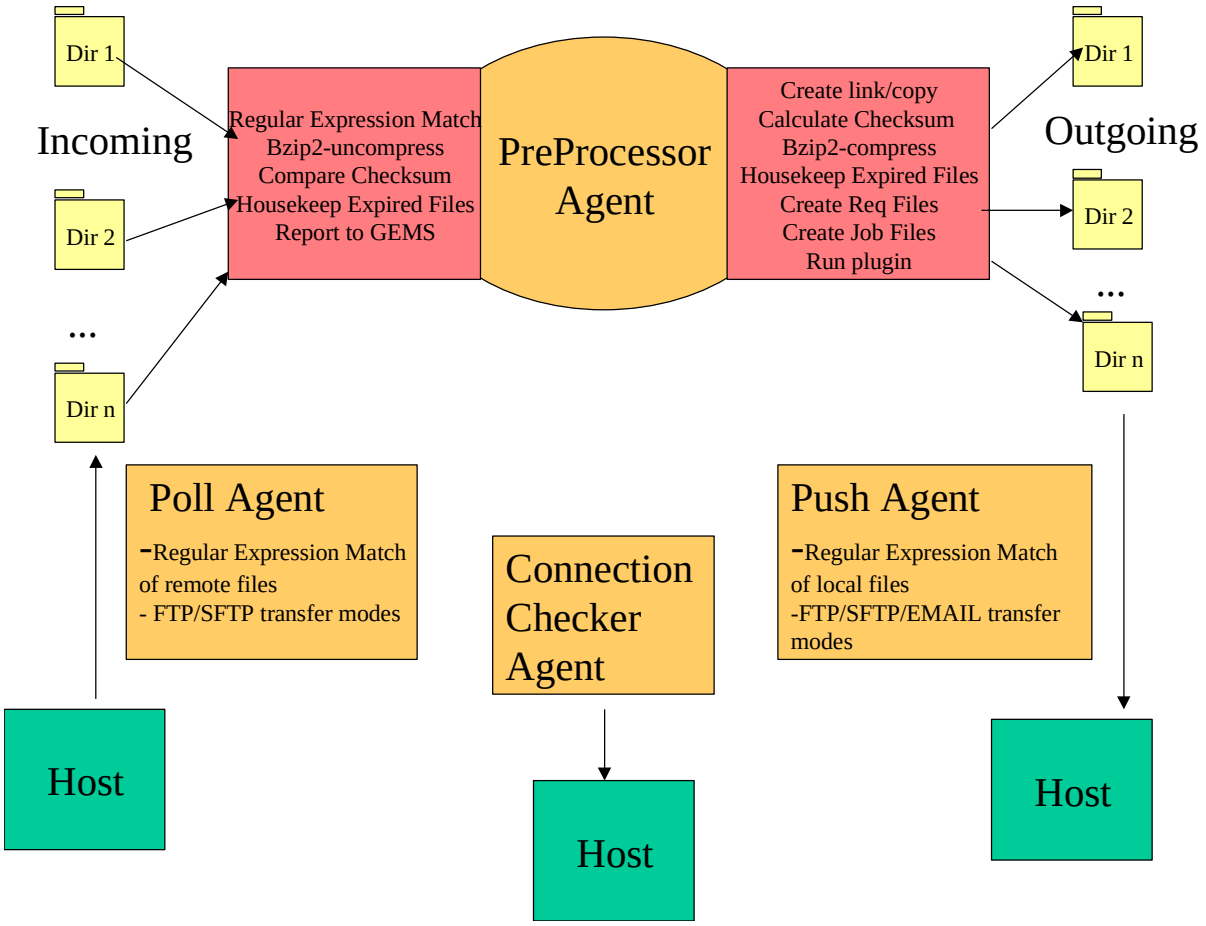
- Transfer files to or from a remote server using various transfer protocols.
- Pre-process files before physically copying or (hard or symbolically) linking them into one or more other directories.
- Check remote connections regularly

The EFTS Agents can run as daemon processes with a configurable cycle time, or as 'single-shot' processes.

Typical uses of the software are the following:

- As central collector of files from several directories to one directory from where the files are further processed.
- As distributor of files from one directory to other directories from where the files are further processed.
- As service on a Windows NT/2000/XP PC copying files and pushing them to another server via a file transfer protocol.
- As automatic distributor and archiver for incoming files.
- As server connection checker.
- As file pull agent that gathers files regularly from remote servers.
- As file push agent that transfers files regularly to remote servers.
- As generator of request files for files that are processed by the EFTS Application. The purpose of the EFTS Application is to provide a secure and reliable file transfer mechanism in an ad-hoc manner, rather than a fixed path approach used by the EFTS Agents.

The EFTS Agents software is written in Java and can therefore be used on multiple platforms without the need of having to recompile the software.



EFTS Agents Schematic

1.3 Agents Overview

The following sections provide an overview of each agent, outlining their possible uses. Precise configuration information can be found in the Configuration section.

1.3.1 Poll Agent

The Poll Agent can be configured to do the following:

- Poll files regularly from a remote host to a local directory.
- Remote files must match the configured regular expression.
- FTP, SecureFTP transfer modes supported.
- Report the last processed file to a status file, independently of GEMS.
- Log messages into a log file.

1.3.2 PreProcessor Agent

The PreProcessor Agent can be configured to do the following:

- Check different incoming directories for new files using regular expression matching of file names.
- Explicitly ignore files matching a regular expression.
- Explicitly ignore files with a certain suffix, e.g. `.TEMP`. This suffix can be defined per incoming directory.
- Report received files to the GEMS utility creating events that can be monitored with the GEMS utility.
- Report the last processed file to a status file, independently of GEMS.
- Copy matched files from an incoming directory to one or more destination directories. The copied files can be hard links, symbolic links or straight copies.
- Call any application (eg. a Plugin), passing the matched file as an argument. Environment variables can be configured that will be available to the application. A current operational use of this feature is detailed below this section.
- Create EFTS request files from templates for the processing of the files with the EFTS.
- Create DVB job files from templates for the processing of the files on the DVB multicast uplink station.
- Check a configurable number of directories for files being processed before creating job files for those data files. That way the job files are not sent out before the data files. A configurable timeout is applied to the creation of job files so that there are not created if the processing of the data files takes too long.
- Generate GEMS events with a configurable facility and process name.
- Sleep for a configurable time after completing one cycle of operation.
- Housekeeping: Delete files from the incoming directory if they are older than a configurable age.
- Housekeeping: Delete files from the outgoing directory if they are still in that directory after a configurable time.
- RMDCN: Create a global sequence number and change file name to contain a prefix, the global sequence number and a suffix. This feature is used for outgoing RMDCN files.
- RMDCN: Create channel sequence numbers (CSNs) and replace the values inside the files with these new CSNs. This feature is used for outgoing RMDCN files.
- Change the time zone internal to the PreProcessor so that all time output is created according to that time zone. This feature is especially useful on the DVB multicast PCs which are distributed over different time zones but all reporting in UTC time.
- Calculate a CRC32-checksum for each file being processed and attach that checksum to the file name. This feature was used on the DVB multicast trial PCs, that way checking the integrity of the received files.
- Compare the CRC32-checksum attached to the file name with the calculated value of the CRC32-checksum and remove the checksum from the file name. This feature was used on the DVB multicast trial PCs, that way checking the integrity of the received files.
- Compress files with the Bzip2 compression method.
- Uncompress files with the Bzip2 compression method.
- Log messages into a PreProcessor log file.
- Housekeeping of log file: Create a new version of the PreProcessor log file after it has reached a configurable size.

1.3.2.1 Preprocessing a matched file with a plugin

The PreProcessor Agent can be configured to call any application (eg. a Plugin), passing the matched file as an argument. Environment variables can be configured that will be available to the application. This approach offers enormous flexibility, allowing existing plugin applications to be reused.

Th plugin feature is currently used operationally whereby a plugin is called to split a matching RECT2LP file into LRIT prologue and data segments for the Visible, Infra Red, and Water Vapour channels.

1.3.3 Push Agent

The Push Agent can be configured to do the following:

- Push files regularly from a local directory to a remote host.
- Local files must match the configured regular expression.
- Files can be sent via FTP, SecureFTP, or emailed.
- Report sent files to the GEMS utility creating events that can be monitored with the GEMS utility.
- Report the last processed file to a status file, independently of GEMS.
- Log messages into a log file

1.3.4 Connection Checker Agent

The Connection Checker Agent can be configured to do the following:

- Check (FTP,SFTP) connections to a remote host by attempting regular logins and log results.
- Report connection problems to GEMS creating events that can be monitored with the GEMS utility

All configurations can be performed using a property file, which is a simple text file that is read at start-up of the agents.

1.4 Agent Execution Modes

Via start-up arguments, all the agents can be run in various execution modes:

- OPERATIONAL: housekeeping and real processing (poll/push/pre-processing) performed.
- TEST: housekeeping and dummy processing (poll/push/pre-processing) performed.
- VALIDATION: same as TEST
- HOUSEKEEPING ONLY: only housekeeping performed
- PROCESSING ONLY: no housekeeping performed.

1.5 Pre-Conditions

The EFTS Agents software requires Java JRE 1.4 or higher. It can run on any platform that supports the Java runtime environment 1.4 or higher.

1.6 Constraints

None

2 OVERVIEW OF SOFTWARE

2.1 Agents

The EFTS Agents comprises the following components:

Component	Task
EFTS_PreProcessor	Detects files using regular expression matching and copies them into one or more destination directories.
EFTS_FtpConnectionChecker	Checks connections to remote hosts regularly using ftp/sftp logins.
EFTS_PushFtpAgent	Pushes files regularly to a remote host.
EFTS_PollFtpAgent	Pulls files regularly from of a remote host.

2.2 Operating Environment

Since the EFTS Agents software is written in Java, it runs on all operating systems with an installed Java Runtime Environment. The class files are contained in Java Archives (JARs) and have to be defined in the CLASSPATH environment variable:

The following library files (which are part of the EFTS Agents distribution) need to be installed on the system, and the jar files included in the CLASSPATH:

Library Name	Contents
EFTS_Agents.jar	Compiled files of the EFTS Agents software.
GEMS.jar	Generic Event Monitoring Software.
NetComponents.jar	Ftp client interface classes.
Javaunix.jar	On Solaris only, native Java classes for system commands. Also requires libjavaunix.so.
Activation.jar	JavaBeans Activation Framework
Commons-logging.jar	Apache Commons Logging framework
Commons-net.jar	Apache Commons Networking framework
j2ssh.jar	SSH Client framework
mail.jar	Java Mail Framework
Wrapper.dll	Java Service Wrapper runtime library (windows use only)
Wrapper.jar	Java Service Wrapper (for NT based services)

3 OPERATIONAL CONFIGURATION ISSSUES

The following describes the issues surrounding the primarily operational configuration for the EFTS Agents.

3.1 Installation

3.1.1 General Information

The EFTS Agents release is packed into a TAR distribution file. From this TAR file, the agents can be installed on any system that has an installed JRE and Windows or X-Windows library. The directory structure of the distribution is:

EFTS_Agents_V2_20 – root directory (version 2.20 in this example)
- **java** – contains the JAR files
- **bin** – startup, shutdown, windows service install/uninstall scripts
- **DataUnits** – sample property files for the agents and windows Java Service Wrapper configuration files
- **termination** – default directory where termination files are stored
- **log** – default logging directory

3.1.2 Installation on a UNIX system

The software should be unpacked in the operations administration account under a Releases directory with a directory name indicating the release of the software. A link called **production** in the administration home account will point to the official release and is updated whenever a new release is made. This link will always point to the latest release and is included in the administration path used when the agents are called.

The configuration of the Agents software is defined in the property files (refer to the Configuration section).

3.1.3 Installation on a Windows system

The directory structure on a Windows system does not allow links to be included in path names. That way it is not possible to link the Releases directory to the latest release. The following environment variables must be set:

Environment Variable	Value
CLASSPATH	all the JAR files identified in section Operating Environment, colon separated
EFTS_AGENTS_HOME	root directory of the EFTS Agents installation, e.g. C:\efts_agents\Releases\EFTS_Agents_V2_20)

On a windows system, the software can be installed as a service that is started at boot time without the need of manual interaction.

To achieve that, batch files are provided enabling the installation of the services.

The following batch files are provided to install the agents as services. They are located in the **bin** directory. The respective agent can be started by starting it from the Windows 'Services' interface, or rebooting the machine.

Batch file	Purpose
InstallEFTS_FtpConnectionChecker-NT.bat	Installs the Connection Checker agent as an NT service.
InstallEFTS_PollFtpAgent-NT.bat	Installs the Poll agent as an NT service.
InstallEFTS_PreProcessor-NT.bat	Installs the PreProcessor agent as an NT service.
InstallEFTS_PushFtpAgent-NT.bat	Installs the Push agent as an NT service.

3.2 The Uninstall Process

On a UNIX platform simply remove the installation directory and reassign the **production** link, if necessary.

On a Windows platform, first uninstall the windows services (if installed during the installation process) by executing the following scripts. Then remove the installation directory:

Batch file	Purpose
UninstallEFTS_FtpConnectionChecker-NT.bat	Uninstalls the Connection Checker agent as an NT service, stopping any running agent first.
UninstallEFTS_PollFtpAgent-NT.bat	Uninstalls the Poll agent as an NT service, stopping any running agent first.
UninstallEFTS_PreProcessor-NT.bat	Uninstalls the PreProcessor agent as an NT service, stopping any running agent first.
UninstallEFTS_PushFtpAgent-NT.bat	Uninstalls the Push agent as an NT service, stopping any running agent first.

3.3 Start Up and Shut Down of the Agents

3.3.1 Command line startup

To start the Agents (on Unix or Windows platforms) execute the following command :

```
java <options> <agent name> <full path to property file>
```

where `<options>` are:

[-help | -version] | [[-versionProp | -vf | -f] [-eo | -ev | -et] [-ehk_only | -ehk_off]]

-help --> print help page
-s --> run agent once only, then terminate
-version --> print the version number of the EFTS Agents distribution
-versionProp --> print the version number of the property file
-vf --> last processed file info (verbose)
-f --> last processed file info (brief)
-eo --> operational execution mode
-ev --> validation execution mode\n
-et --> test execution mode\n
-ehk_only --> run housekeeping only execution mode
-ehk_off --> housekeeping disabled execution mode

where `<agent name>` is one of :

```
EFTS_FtpConnectionChecker
EFTS_PollFtpAgent
EFTS_PreProcessor
EFTS_PushFtpAgent
```

defining the agent to be started.

The agents can be stopped manually by putting a termination file in the termination directory that is defined in the property file. The contents of the termination file is irrelevant, it will be deleted when the process shuts down.

3.3.2 Startup scripts on Windows Systems

On a Windows platform, if the processes are not to be run as services they can be started from a BATCH script as follows:

Batch file	Purpose
RunEFTS_FtpConnectionChecker.bat	Starts the Connection Checker agent using the property file defined in the batch script.
RunEFTS_PollFtpAgent.bat	Starts the Poll agent using the property file defined in the batch script
RunEFTS_PreProcessor.bat	Starts the PreProcessor agent using the property file defined in the batch script
RunEFTS_PushFtpAgent.bat	Starts the Push agent using the property file defined in the batch script.

They can be stopped manually by putting a termination file in the termination directory that is defined in the property file. The contents of the termination file is irrelevant, it will be deleted when the process shuts down.

3.4 Configuration

All Agents are configured by properties defined in a property file. Each agent has its local set of properties; these will be described in the subsequent sections.

The following global set of properties is applicable to all agents:

Property family	Meaning	
Logging	The verbosity of the process log messages and the information to be supplied in the log messages (like package and thread info), as well as name of log file and path.	
	LogSetting1	The agent logging level. One or more of: ALL,WARN, DEFAULT.
	LoggingThreadInfo	True or false
	LoggingPackageInfo	True or false
	LoggingTimeInfo	True or false
	LogPath	Full pathname to the directory where any log files will be appended
	LogFile	Leaf filename of the agents's log file.
	StatusFile	Leaf name (in the LogPath directory) of the last processed file.
UseGEMSReporting	Whether the agent should log GEMS events: Yes or No	
TimeZone	Time zone of the log messages being created, usually set to GMT+0, automatically calculating the offset from the time zone that is set on the host.	
HouseKeeping	housekeeping interval and log file sizes	
	HousekeepingSize	Max size in bytes of the agent's log file. Once this size is reached the current one is archived and a new one is created.
	HousekeepingInterval	The number of processing cycles before housekeeping is performed.
Termination		
	Directory	Directory where the processes look for termination files to stop processing gracefully.
	File	Termination filename, default is the agent name.
Mission	Key that identifies the agents. Used as a prefix in any GEMS event raised by the agent.	
propfile.version	The version of the property file.	

3.4.1 Housekeeping

Data can accumulate on the disks. The agents can automatically delete files that are older than a configured age:

- PreProcessor agent: Delete files in the incoming or copy to directories. If symbolic linking is being used, the PreProcessor also checks that the source (linked to) file actually exists - if not any links to the source file will be deleted.
- This housekeeping can be configured to take place every C cycles. No housekeeping is performed on remote servers.

3.4.2 Properties for the PreProcessor Agent

The PreProcessor configuration properties are listed in the following table:

Property	Meaning
EFTS.PreProcessor.CreateReqFile	(optional) Flag to be set to "Yes" if an EFTS request file shall be created for each incoming file. Used for processing of the files by the EFTS.
EFTS.PreProcessor.CreateJobFile	(optional) Flag to be set to "Yes" if a job file shall be created for each incoming file. Used for processing of the files on the dvb uplink server.
EFTS.PreProcessor.AddChecksumToFileName	(optional) Flag to be set to "Yes" if a CRC32-checksum shall be calculated for each incoming file and attached to the file name. Not currently used.
EFTS.PreProcessor.GetChecksumFromFileName	(optional) Flag to be set to "Yes" if a CRC32-checksum shall be calculated for each incoming file and compared to the CRC32-checksum attached to the file name. Not currently used.
EFTS.PreProcessor.CompressBZip2	Flag to be set to "Yes" if a file shall be compressed using the Bzip2-compression. Not currently used.
EFTS.PreProcessor.UncompressBZip2	Flag to be set to "Yes" if a file shall be decompressed using the Bzip2-compression. Not currently used.
EFTS.PreProcessor.CreateHardLinks	On Solaris-Systems only: Flag to be set to "Yes" if a hardlink to the source file shall be created instead of copying the file. If set to “No” symbolic links will be used, but the IncomingN.MoveToFirstDir must be specified. On Windows platforms straight copies will be performed.
EFTS.PreProcessor.GEMS.Facility	GEMS reporting property: facility name to be reported.
EFTS.PreProcessor.GEMS.Process	GEMS reporting property: process name to be reported.
EFTS.PreProcessor.Server.CheckInterval	Interval in msec of checking incoming directory for new files.
EFTS.PreProcessor.Error.Dir	Error directory for files that cannot be processed.
EFTS.PreProcessor.DataDwellTime	(optional) To be used if CreateJobFile is set to Yes only: Time in msec to wait for completion of data file transfer before a job file will be created.
EFTS.PreProcessor.DataDwellDirectory	(optional) To be used if CreateJobFile is set to Yes only: Directories where to look for files being transferred by the EFTS.
EFTS.PreProcessor.DataDwellMaxRetries	(optional) To be used if CreateJobFile is set to Yes only: Maximum number of retries after which the job files will not be created anymore.
EFTS.PreProcessor.Incoming.HousekeepingHours	Files in all incoming directories will be deleted if older than this (hours) value.
EFTS.PreProcessor.Incoming.HousekeepingMillisecs	Files in all incoming directories will be deleted if older than this (milliseconds) value. Overrides the Hours setting.

Property	Meaning
The following properties are indexed by an integral value (N) after the Incoming node, referred to as a PreProcessor agent incoming 'block'. There is no limit to the number of blocks although it is considered good practice to keep the number small. Each 'block' is processed sequentially.	
EFTS.PreProcessor.IncomingN.Dir	Directory to check for new file arrivals.
EFTS.PreProcessor.IncomingN.MoveToFirstDir	(optional, but mandatory if CreateHardLinks is set to "No", implying symbolic links on Unix platforms). Directory that matched files in the incoming directory will be moved to before any copyTo handling is performed.
EFTS.PreProcessor.IncomingN.CopyToM	Directory where to copy files to. There can be M copy to directiories, where M is an integer >0.
EFTS.PreProcessor.IncomingN.CopyToM.Plugin.Name	(optional) Path to plugin application which will be called with 1 argument: the full pathname to the matched file.
EFTS.PreProcessor.IncomingN.CopyToM.Plugin.EnvVarX.name EFTS.PreProcessor.IncomingN.CopyToM.Plugin.EnvVarX.value	(optional). Defines any environment variable name/value pairs for the Plugin specified by Plugin.Name. There can be X pairs, where X is an integer >0.
EFTS.PreProcessor.IncomingN.FileMatchRegexp	Files to be copied have to match this regular expression. Needs to be configured for each incoming directory to be checked.
EFTS.PreProcessor.IncomingN.FileExcludeRegexp	(optional) Files to be copied must not match this regular expression. Takes precedence over the FileMatchRegexp.
EFTS.PreProcessor.IncomingN.CopyTo.ReqFileTempl	(optional) Request file template to be used for EFTS request file generation. Can be set for each CopyTo directory.
EFTS.PreProcessor.IncomingN.CopyTo.JobFileTempl	(optional) Job file template to be used for transfers to dvb uplink station. Can be set for each CopyTo directory.
EFTS.PreProcessor.IncomingN.CopyTo.JobFileMaxNameLength	(optional) Numerical value to limit the maximum file length of job file names. Can be set for each CopyTo directory.
EFTS.PreProcessor.IncomingN.CopyTo.JobReqFileTempl	(optional) EFTS request file template to be used for job files. Can be set for each CopyTo directory.
EFTS.PreProcessor.IncomingN.IgnoreSuffix	File extension to be ignored in incoming directory.

3.4.2.1 Configuration as an RMDCN-PreProcessor

The RMDCN-PreProcessor regularly copies files from an incoming directory to several outgoing directories on the local host. It also changes the file names to follow a global sequence and the internal "channel sequence number" to follow

In order to configure the PreProcessor the following properties have to be set in addition to the common properties described in 3.4.2:

Property	Meaning
EFTS.PreProcessor.CreateGlobalSequenceNumber	Flag to be set to "Yes" if a global sequence number in the renamed file names should be created.
EFTS.PreProcessor.GlobalSequenceNumber.FilePrefix	File prefix to be used in renamed file names.
EFTS.PreProcessor.GlobalSequenceNumber.FileSuffix	File suffix to be used in renamed file names.
EFTS.PreProcessor.CreateCSNSequenceNumber	Flag to be set to "Yes" if a channel sequence number inside the files should be created.

3.4.3 Configuration as a PushFtpAgent

The PushFtpAgent regularly transfers files to a server.
In order to configure the PushFtpAgent the following properties have to be set:

Property	Meaning
FtpAgents.PushFtpAgent.Facility	GEMS reporting property: facility to be reported.
FtpAgents.PushFtpAgent.Process	GEMS reporting property: process name to be reported.
FtpAgents.PushFtpAgent.CheckInterval	Interval of pushing of files, in msec.
FtpAgents.PushFtpAgent.NoTransfer.Extension	Extension of files not to be transferred.
The following properties are indexed by an integral suffix (N), referred to as a Push agent 'block'. There is no limit to the number of blocks although it is considered good practice to keep the number small. Each 'block' is executed as a separate Java Thread.	
FtpAgents.PushFtpAgent.TransferMethodN	(optional, default ftp) The transfer method used. Options are ftp, sftp, or mail.
FtpAgents.PushFtpAgent.RegularExpression.MatchN	Files to be transferred have to match this regular expression. Needs to be configured for each local directory to be checked.
FtpAgents.PushFtpAgent.RegularExpression.Exclude N	(optional) Files to be transferred must not match this regular expression. Takes precedence over the Match value.
FtpAgents.PushFtpAgent.IpAddressN	IP address of server (SMTP address if transfer method is 'mail'. Needs to be configured for each server.
FtpAgents.PushFtpAgent.UserN	User name of server login. Needs to be configured for each server.
FtpAgents.PushFtpAgent.PasswordN	Password of server login. Needs to be configured for each server.
FtpAgents.PushFtpAgent.ToAddressN	(mandatory if TransferMethod is 'mail'). The email address the matching file will be sent to.
FtpAgents.PushFtpAgent.FromAddressN	(mandatory if TransferMethod is 'mail'). The email address the matching file will be sent from.
FtpAgents.PushFtpAgent.SendMailAsAttachment1	(only if TransferMethod is 'mail'). (optional, default true). If true the matching file will be emailed as an attachment, otherwise the file will be sent within the body of the email message (if the SMTP server is able to include the file type in the body).
FtpAgents.PushFtpAgent.ConnectionTimeoutN	Connection timeout in msec for connection attempts to server. Needs to be configured for each server.
FtpAgents.PushFtpAgent.ConnectionRetriesN	Number of connection retries for connection attempts to server. Needs to be configured for each server.
FtpAgents.PushFtpAgent.ConnRetryTimeIntervalN	Retry interval in msec for connection retries to server. Needs to be configured for each server.
FtpAgents.PushFtpAgent.FileTimeoutN	Timeout in msec for file transfer attempts to server. Needs to be configured for each server.
FtpAgents.PushFtpAgent.FileRetriesN	Number of retries for file transfer attempts. Needs to be configured for each server.
FtpAgents.PushFtpAgent.FileRetryTimeIntervalN	Time in msec difference between two retry attempts. Needs to be configured for each server.
FtpAgents.PushFtpAgent.FileMaxLifeTimeN	Any files in the local directory will be deleted if older than this time (in msec).
FtpAgents.PushFtpAgent.FileAgeAlarmLimitN	(optional, default no limit, no alarms will be raised). If any file in the local directory is older than this time (in msec) it will be marked as reaching its alarm limit.

Property	Meaning
FtpAgents.PushFtpAgent.FileAgeAlarmThresholdN	(optional, default 30) Alarm threshold for number of files in the local directory being older than a certain age (see AgeAlarmLimitN). If reached an ALARM will be raised.
FtpAgents.PushFtpAgent.RemoteDirN	Directory on server where the files shall be transferred to.
FtpAgents.PushFtpAgent.LocalDirN	Directory on local server where the files shall be transferred from.
FtpAgents.PushFtpAgent.DeleteLocalFilesN	(optional, default false) If "true" local files will be deleted after they are sent.

3.4.4 Configuration as PollFtpAgent

The PollFtpAgent regularly transfers files from an ftp server to the local host.
In order to configure the PollFtpAgent the following properties have to be set:

Property	Meaning
FtpAgents.PollFtpAgent.Facility	GEMS reporting property: facility to be reported.
FtpAgents.PollFtpAgent.Process	GEMS reporting property: process name to be reported.
FtpAgents.PollFtpAgent.CheckInterval	Interval of ftp polling of files.
FtpAgents.PollFtpAgent.NoTransfer.Extension	Extension of files not to be polled.
The following properties are indexed by an integral suffix (N), referred to as a Poll agent 'block'. There is no limit to the number of blocks although it is considered good practice to keep the number small. Each 'block' is executed as a separate Java Thread.	
FtpAgents.PollFtpAgent.TransferMethodN	(optional, default ftp) The transfer method used. Options are ftp, or sftp.
FtpAgents.PollFtpAgent.RegularExpression.MatchN	Files to be polled have to match this regular expression. Needs to be configured for each local directory to be checked.
FtpAgents.PollFtpAgent.RegularExpression.ExcludeN	(optional) Files to be received must not match this regular expression. Takes precedence over the Match value.
FtpAgents.PollFtpAgent.IpAddressN	IP address of ftp server. Needs to be configured for each ftp server.
FtpAgents.PollFtpAgent.UserN	User name at ftp server to be checked. Needs to be configured for each ftp server.
FtpAgents.PollFtpAgent.PasswordN	Password at ftp server to be checked. Needs to be configured for each ftp server.
FtpAgents.PollFtpAgent.ConnectionTimeoutN	Connection timeout in msec for connection attempts to ftp server. Needs to be configured for each ftp server.
FtpAgents.PollFtpAgent.ConnectionRetriesN	Number of connection retries for connection attempts to ftp server. Needs to be configured for each ftp server.
FtpAgents.PollFtpAgent.ConnRetryTimeIntervalN	Number of connection retries for connection attempts to ftp server. Needs to be configured for each ftp server.
FtpAgents.PollFtpAgent.FileTimeoutN	Timeout in msec for file transfer attempts to ftp server. Needs to be configured for each ftp server.
FtpAgents.PollFtpAgent.FileRetriesN	Number of retries for file transfer attempts. Needs to be configured for each ftp server.
FtpAgents.PollFtpAgent.FileRetryTimeIntervalN	Time difference in msec between two retry attempts. Needs to be configured for each ftp server.
FtpAgents.PollFtpAgent.RemoteDirN	Directory on ftp server where the files shall be transferred from.

Property	Meaning
FtpAgents.PollFtpAgent.LocalDirN	Directory on local server where the files shall be transferred to.
FtpAgents.PollFtpAgent.DeleteRemoteFilesN	(optional, default false) If "true" remote files will be deleted after they are retrieved.

3.5 Testing and Future Developments

Future development include:

- Push Agent: add an optional configuration property allowing the ability to execute push 'blocks' sequentially, rather than in separate threads. This would make the file delivery sequence more predictable and less dependent on transfer times.
- Push Agent: add an optional configuration property allowing the ability to execute individual transfers (**within** a push 'block') sequentially, rather than in separate threads. This would make the file delivery sequence more predictable and less dependent on transfer times.