

**Note**

Before using this information and the product it supports, read the information in “Notices” at the end of this document.

This edition applies to version 4.0.1.6 of IBM FileNet System Monitor (product number 5724-R91) and to all subsequent releases and modifications until otherwise indicated in new editions.

**Copyright © 2000-2010 CENIT AG, Germany; Copyright © 2005, 2010 IBM Corp., USA**  
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

# Table of Contents

<b>Preface</b> .....	<b>3</b>
About this document.....	3
<b>FSM CALA</b> .....	<b>5</b>
Who should read this chapter?.....	5
CALA installation.....	6
Uninstall CALA.....	11
<b>Component Architecture</b> .....	<b>12</b>
CALA System platforms.....	12
Supported JAVA JRE or JDK versions (CALAGUI and CALA V2S Editor prerequisite).....	13
Implementation on Microsoft Windows based systems.....	14
Configuration file logctlsrv.conf for a Windows service installation.....	15
Client / Server Architecture.....	16
Implemented components.....	17
Possible component architecture (predecessors / successors).....	23
Communication between CALA components.....	25
Default TCP ports used by CALA components.....	26
Event caching.....	27
<b>CALA configuration</b> .....	<b>28</b>
Global information.....	28
Global configuration files.....	29
Datatype ibm_cm8_eventlog - IBM CM8 Eventlog.....	30
Datatype ibm_cmcs_archive - IBM CS archive log.....	32
Datatype ibm_cmcs_error - IBM CM8 agent and CS server error log.....	34
Datatype ibm_cmcs_retrieve - IBM CS retrieve log.....	36
Datatype ibm_cmod_log - IBM CM On Demand Logfile.....	37
Datatype ibm_cmrm_asyncr - IBM CM RM Asyncr logfiles.....	38
Datatype ibm_cmrm_migrator - IBM CM RM Migrator Logfiles.....	39
Customization Issues.....	40
Sample configuration.....	44
<b>Local Environment File fsm4ibm_srv_env.sh</b> .....	<b>45</b>
<b>Copyright notice</b> .....	<b>51</b>
IBM FileNet System Monitor (18 March 2011) .....	51

# Preface

## About this document

### *Who should read this guide?*

The target audience for this guide are those who install or maintain FSM environments.

Every effort has been made to provide you with complete installation instructions. If information becomes available after the creation of the installation media from which you accessed this guide, we will provide an updated version of the guide on the IBM/FileNet Customer Service and Support web site (<http://www.ibm.com/support>). As a general rule, you should refer to the IBM web site to obtain the current version of this guide.

This guide provides instructions for installing and/or upgrading IBM FileNet System Monitor, and identifies the IBM/FileNet and 3rd Party products that are certified for the current release. Be aware that each release of IBM FileNet System Monitor may have multiple Interims Fixes, or Fix Packs available for installation, each with potentially different dependencies and installation requirements. Therefore, before you attempt to install or upgrade IBM FileNet System Monitor, review the list of releases and their associated dependencies on the IBM Support web site (<http://www.ibm.com/support>).

### *Before you start*

Users of the guide should have knowledge about Unix and/or Microsoft Windows® operating system, web servers, database systems and middleware platforms. The configuration of managed systems (clients) requires advanced knowledge of all IBM ECM systems that should be monitored.

If you lack the requisite skill sets it is strongly recommended to have IBM Lab Services or a certified ValueNet Partner install this product.

### *Where you find this guide*

You can find this documentation on the FSM installation media in the following folder:

UNIX: `<Mount point>/INSTALL/docs`

Windows: `<Drive letter>:\INSTALL\docs`

## ***Feedback on documentation***

Send your comments by e-mail to [comments@us.ibm.com](mailto:comments@us.ibm.com). Be sure to include the name of the product, the version number of the product, and the name and part number of the book (if applicable). If you are commenting on specific text, include the location of the text (for example, a chapter and section title, a table number, a page number, or a help topic title).

## FSM CALA

This chapter provides detailed installation information about the FSM CALA.

### **Who should read this chapter?**

We recommend this chapter for all who are interested in adapter details or in tuning the adapter.

# CALA installation

## Post Distribution: CALA Installation

This chapter contains a detailed description of the installation script **calainst.sh**. This script is used by installation via the CALA installer used for FSM clients and servers..

The settings for the final installation directory `CALA_DIR` and the CALA cache directory `CALA_CACHE_DIR` can be specified in the installer GUI.

The table shows the default values:

Installation context	Settings
UNIX-based FSM client or server	<code>CALA_DIR = /opt/FileNet/SysMon/cala</code> <code>CALA_CACHE_DIR = /opt/FileNet/SysMon/cala/.calacache</code>
Windows-based FSM client or server	<code>CALA_DIR = C:/Program Files/FileNet/SysMon\cala</code> <code>CALA_CACHE_DIR = C:/Program Files/FileNet/SysMon\cala \.calacache</code>

These directories are created if they do not exist yet. If you have already distributed CALA (via the FSM installer), the installations share the same directories. In this case, you always have only one instance of CALA, the configuration entries are merged into one configuration file.

The script checks if there is sufficient disk space in the filesystem where the CALA is to be installed.

The next step is to stop the adapter if it is already installed. The script calls the scripts and binaries installed by the standard installation to stop CALA. If CALA was installed manually using different script names it must be stopped manually before trying to distribute again.

The standard calls for stopping the CALA are as follows:

interpreter type	CALAstop command
aix4-r1	execute <code>\$CALA_DIR/cala.sh</code> with parameter <code>shutdown</code>
hpux10	execute <code>\$CALA_DIR/cala.sh</code> with parameter <code>shutdown</code>
solaris2	execute <code>\$CALA_DIR/cala.sh</code> with parameter <code>shutdown</code>
w32-ix86	execute <code>\$CALA_DIR/logctlcmd.exe</code> with parameter <code>shutdown</code> or if CALA is running as service execute <b>net stop cala_srv</b>

The CALA configuration file is now generated from the `.cala` input files found in the `repos` subdirectory of the CALA installation directory. For a detailed description of the merging process see the *Technical details of CALA configuration*.

## Installation of binaries

The binaries have already been transferred to the client in a previous step.

There are two types of installation, Client or Server, that differ in the binaries that are activated:

### *Client installation:*

- ascfileread
- tecfmtfilt
- v2fmtfilt
- oracleread
- ntevtlogread (Windows only)
- mssqlread (Windows only)
- calamon

### *Server installation*

- snmpread
- msgclsfsrv
- tecfmtemit
- tecifcsrv
- snmpemit
- smtpemit
- cmdemit
- reportemit
- mysqlemit

The Server installation contains all client components as well.

## Installed binaries

The following table contains a list of binaries that can be found in the CALA installation directory after a CALA configuration has been installed.

Name	Platform	Function
ascfileread[.exe]		reader for ASCII logfiles
cala_srv.exe	w32-ix86	FSM CALA service binary
calamon[.exe]		monitoring engine



Name	Platform	Function
calaproxy[.exe]		proxy for communication over firewall
cmdemit[.exe]		command emitter
libcala.dll	w32-ix86	library for common CALA functions
libcala.sl	hpux10	library for common CALA functions
libcala.so	aix4-r1, solaris2	library for common CALA functions
logctcmd[.exe]		cli interface
logctsrv[.exe]		main CALA control process
msgclsfsrv[.exe]		message classification server component
mssqlread[.exe]	w32-ix86	reader for MSSQL databases
mysqlemit[.exe]		writer for MySQL databases
oracleread[.exe]		reader for Oracle databases
reportemit[.exe]		report emitter
smtpemit		smtp emitter
snmpemit		snmp emitter
snmpread		snmp reader
tecfmtegit[.exe]		emitter for TEC formatted events
tecfmtfilt[.exe]		filter for TEC format
tecifcsrvend[.exe]		Endpoint TEC interface server (Tivoli based communication)
tecifcsrvsec[.exe]		secure TEC interface server (Tivoli based communication)
tecifcsrvuns[.exe]		unsecure TEC interface server (TCP/IP based communication)
v2fmtfilt[.exe]		filter for V2S format
*msg.dll	w32-ix86	additional DLLs for logging

## Configuration of autostart

After successful installation of the binaries, the CALA is configured for autostart. This depends on the target system.

interpreter type	autostart configuration
aix4-r1	<ul style="list-style-type: none"> <li>generate start script <code>/etc/rc.cala</code></li> <li>appended the following entry to <code>/etc/inittab</code> if it does not exist: <code>cala:2:once:/etc/rc.cala</code></li> </ul>
hpux10	<ul style="list-style-type: none"> <li>generate start script <code>/sbin/init.d/cala</code></li> <li>create link <code>/sbin/rc3.d/S500CenitCALA</code></li> <li>create link <code>/sbin/rc0.d/K500CenitCALA</code></li> </ul>
solaris2	<ul style="list-style-type: none"> <li>generate start script <code>/etc/init.d/cala</code></li> </ul>

interpreter type	autostart configuration
	<ul style="list-style-type: none"> <li>create link <code>/etc/rc3.d/S500CenitCALA</code></li> <li>create link <code>/etc/rc0.d/K500CenitCALA</code></li> </ul>
w32-ix86	<ul style="list-style-type: none"> <li>execute <code>cala_srv -remove</code> to remove CALA service if it was already installed</li> <li>execute <code>cala_srv -auto</code> to install service</li> </ul>

## Registry entries on Windows

The installation of the FSM CALA service creates the following registry key:

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\cala\_srv

The following subkeys are added by the installation script calainst.sh:

Subkey	Meaning
CALA_DIR	CALA installation directory
CALA_CACHE_DIR	Directory for CALA cache files
TIV_ENV_FILE	Full-qualified name of the Tivoli environment file for DOS (setup_env.cmd) This entry is needed by the Tivoli based versions of the TEC Interface Server only (Secure or Endpoint version).
CALA_ENV_FILE	Full-qualified name of the additional CALA environment file for DOS (usually %CALA_DIR%/cala_env.cmd; see <i>Customization Issues</i> subchapter <i>Adding environment settings for CALA</i> for details).

These registry keys can be overridden by specifying environment variables.

CALA will now be started.

interpreter type	CALA start command
aix4-r1	execute <code>\$CALA_DIR/cala.sh</code> with parameter <code>startup</code>
hpux10	execute <code>\$CALA_DIR/cala.sh</code> with parameter <code>startup</code>
solaris2	execute <code>\$CALA_DIR/cala.sh</code> with parameter <code>startup</code>
w32-ix86	execute <code>net start cala_srv</code>

## Processes

The following processes can be found in the process list (`ps -ef` on UNIX, Taskmanager on Windows) if CALA is up and running.

- ascfileread

- cala\_srv
- calamon
- cmdemit
- logctlsrv
- msgclsfsrv
- mssqlread
- mysqlemit
- ntevtlogread
- oracleread
- reportemit
- snmpread
- snmpemit
- smtpemit
- tecfmtemit
- tecfmtfilt
- tecifcsrv[end|sec|uns]
- v2fmtfilt

**NOTE** Only components referenced in the configuration file logctlsrv.conf will be started.

## Distribution logfiles

After successful distribution and installation of CALA, all temporary files will be removed.

The distribution logfile `caladist.log` can be found in the CALA installation directory. The contents of the file will be shown in a subwindow of the CALA installer during the installation process.

## Possible error conditions

- Not enough space for CALA binaries in `<directory>`. This message can occur if CALA is already installed and there is only little disk space left. The script does not check if any binaries are overwritten during distribution but requires as much space as if it was performing a new installation.
- Windows service `FSM CALA` could not be installed. If you specified a user for CALA installation, this error message can indicate that the user and / or password is invalid.

All other error texts are self-explaining.

# Uninstall CALA

## Uninstall using the Installer GUI

To remove CALA from a client, you can use the installer GUI. Check the **Remove** option and start the installation process. In this case the CALA installation script **calainst.sh** is called with the command line parameter `-r`.

**CAUTION** This will remove the COMPLETE configuration! Do not use this option to remove specific logfiles or monitors from the CALA configuration.

## Manual uninstall

If you do not want to use the installer GUI, you must remove CALA manually.

If CALA is installed in the default directory described above, you can execute the installation script **calainst.sh** with the command line parameter `-r` to remove CALA and all related files.

**NOTE** On Windows platforms, the file `calainst.sh` cannot be deleted because it is locked during execution. The same applies to the installation directory `CALA_DIR`. The script and the directory must be removed manually.

If the script **calainst.sh** is not available or CALA is installed in another directory, you must remove the installation manually. The required actions are described in the following paragraphs.

First, make sure to stop CALA by issuing the appropriate stop command. The table of interpreter types and corresponding stop commands can be found above.

Additional de-installation steps depending on the interpreter type:

*w32-ix86*

Remove the CALA service by executing **cala\_srv -remove** from the command line. This removes the service and all related entries from the registry.

*hpux10, solaris2*

Remove the links mentioned in the table *auto start configuration* and the start script.

*aix4-r1*

Remove the `/etc/inittab` CALA-entry mentioned in the table *auto start configuration* and the start script.

Finally, remove the CALA installation directory and the CALA cache directory. For the location of these directories see table at the start of this chapter.

## Component Architecture

CALA is realized as a multi component Client/Server architecture, which enables customers to realize any kind of centralized and distributed Logfile and monitoring architecture. Almost all components are available on a comprehensive list of platforms (see restrictions on below site).

### **CALA System platforms**

For detailed information about supported server and client platforms check the latest release notes.

## Supported JAVA JRE or JDK versions (CALAGUI and CALA V2S Editor prerequisite)

For detailed information about required Java JRE or JDK versions for JAVA tools check latest release notes.

## Implementation on Microsoft Windows based systems

Implementation of CALA on the Windows system platform has been implemented as an Windows Service.

### *CALA installation as Windows Service*

Installation of the Windows Service is performed using the program **cala\_srv.exe**.

Installation with start mode manual : **cala\_srv.exe -install**

Installation with start mode automatic : **cala\_srv.exe -auto**

The Windows Service normally runs under the Local System Account. To install the service as another user, use the following commands:

Installation with start mode manual : **cala\_srv.exe -install <user> <password>**

Installation with start mode automatic : **cala\_srv.exe -auto <user> <password>**

### *CALA de-installation on Windows systems*

To remove the CALA Windows service start **cala\_srv.exe -remove** from the command line.

## Configuration file logctlsrv.conf for a Windows service installation

If CALA is installed as an Windows service, configuration file **logctlsrv.conf** must either be placed in directory/folder `%SystemRoot%\system32\config` or in a directory/folder of your choice, which is mapped to the environment variable `CALA_DIR` of the Windows system environment.

The CALA Windows service reads environment variables `CALA_DIR` and `CALA_CACHE_DIR` out of the registry (registry key `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cala_srv`) if they are not mapped in the environment.

If all registry keys are set properly (see chapter Tivoli integration, Post distribution: CALA installation for details) there is no need to reboot the Windows system.

**NOTE** The CALA processes can also be started as a normal program instead of an Windows service. In this case the CLI-program **logctcmd** (refer to description) should be used should be used for starting and stopping of the CALA components.

**NOTE** On Windows, CALA needs the file **PSAPI.DLL** to be available on the system. The file is automatically installed with CALA in the CALA installation directory and must not be removed.



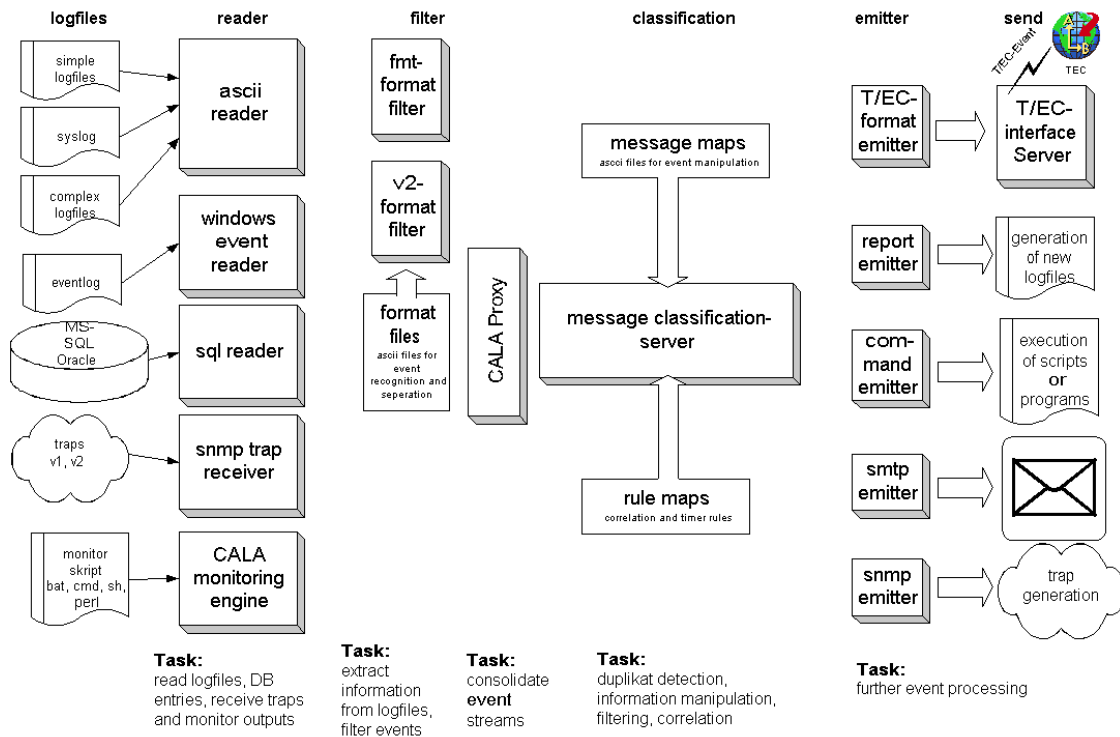
## Client / Server Architecture

To satisfy various requirements (source-independent duplicate recognition, performance, configuration during operation), a specific client/server architecture was developed for the CALA system which distinguishes between the following functions:

- Reading of event sources
- Event filtering
- Classification and duplicate recognition
- Transmitting data (e.g. sending events to the T/EC)

All component of this architecture can be implemented on various systems, or just on a single system (computer).

This diagram illustrates a possible architecture of the CALA system:



This open architecture enables CALA to be implemented at almost any desired level of complexity or heterogeneity.

In addition, the CALA firewall component calaproxy can be interposed between any FIR-based component.

## Implemented components

### *Read-only component (Reader)*

Readers can be used to read event sources. Event sources can be regular files (Logfiles) or pipes, but also Windows Event Logs. The following readers are components of the CALA module:

Component name	Component type	Description
ascfileread	ASCII File reader	This component reads files and pipes available in ASCII format.
ntevtlogread	Windows Event Log reader	This component reads out the Windows Event Log.
mssqlread	MS SQL database reader	This component reads logfiles written into a MSSQL database.
oracleread	Oracle database reader	This component reads logfiles written into a oracle database.
jdbcread	JDBC database reader	This component reads logfiles written into a database accessible via JDBC.

For all readers there is a special CLI parameter (-E), which should be used whenever the intention is solely to process events from the sources being read and which have been written to (generated) since the adapter was started.

**NOTE** As an option, other readers can be implemented on a customer-specific basis.

### *Filter component (Filter)*

Filters are used to disassemble data streams, which were read out of event sources by readers. In all cases, readers are only able to transmit to filters because only they have the ability to disassemble unstructured data streams into events (FIRs) based on format definitions (format files).

In technical terms, filters are arranged between the reader and the processing process or the emitter process.

Component name	Component type	Description
tecfmfiltr	T/EC Format filter	Interprets and classifies input from components ascfileread and ntevtlogread based on Tivoli .fmt files.  This protects existing customer investment in format files.

Component name	Component type	Description
v2fmtfilt	Complex filter	This component filters and interprets input from reader ascfileread based on the CALA-format description.

## ***Event generating components***

Component name	Component type	Description
snmpread	SNMP trap receiver	Receives SNMP traps and forwards them as CALA events (FIRs) to the specified targets.
calamon	Monitoring engine	Executes monitoring scripts/programs and generates events (FIRs) depending on return codes or output.

## ***Processing server msgclsfsrv (Message Classification Server)***

Correlation system msgclsfsrv is the brain behind CALA and is used for duplicate detection, event handling and computing (basic forms of computing).

Event handling includes the functions of event suppression as well as escalation (change in severity).

Component name	Component type	Description
msgclsfsrv	Classification server	This component is the brain behind CALA. It contains the functions of classification, duplicate detection, event suppression and escalation.

## ***Sub components Rules Engine and Message Mapping (msgclsfsrv sub components)***

The rules engine and the message mapping component are subcomponents of the message classification server and are used to process correlating events and timer on events as well as any manipulation on events.

For detailed information about the Rules Engine and the Message Mapping Component refer to related sub chapters within this chapter.

## Sub component Completer (msgclsfsrv sub component)

The sub component Completer in process msgclsfsrv is used for downstream (i.e. after processing by the central processing server) bulk setting or deleting of slots (Tivoli T/EC slots) as a function of other existing or unmapped slots, or to fade out slots.

Language constructs such as *if!*, *unless!* or *for!* are implemented for processing purposes based on the existence (if) or absence (unless) of slots.

Application area:

Mapping of unmapped default slots, e.g. severity.

Fading out of mapped slots, e.g. those required for internal processing.

## Sub component Remapper (msgclsfsrv sub component)

Sub component Remapper in process msgclsfsrv is capable of re-mapping slot contents, or of defining new slots.

This enables the system to rename event class names in a customer-specific manner.

Example:

Changing LogfileBase to <customername>\_LogfileBase.

Comment: the event class <customername>\_LogfileBase must be defined in a BAROC file.

## Emitter components

Emitters (Senders) are used for sending or subsequently processing events, e.g. after a report has been raised. The following emitters/senders are available:

Component name	Component type	Description
tecfmtemit	T/EC-Event-Preparation	This component prepares the logs to be processed and sends them to the T/EC transmit component (see below).
cmdemit	Task Engine	This component is capable of executing any commands. Parameters are read out of Message Map files.
reportemit	Reporting Emitter	This component is used for all the events. Events may be reported in T/EC Event dump format or in a own format specified in a template file.
snmpemit	SNMP Trap Emitter	This components throws SNMP events from received CALA FIRs.
smtpemit	SMTP Emitter	This component sends emails to report the received CALA FIRs.

Component name	Component type	Description
jdbcemit	Database Emitter	This component writes events to a database accessible via JDBC.

### *T/EC transmit component*

Component name	Component type	Description
tecifcsrv	T/EC Interface Server	<p>This component sends prepared events to the T/EC. There are 3 variants of this component:</p> <ul style="list-style-type: none"> <li>Secure Version: oserv communication (ManagedNode)</li> <li>Unsecure Version: TCP/IP communication (EIF)</li> <li>Endpoint Version: Tivoli TMA communication</li> </ul>

### *Application proxy for DMZ*

Component name	Component type	Description
calaproxy	Application proxy	This component is used as an application proxy in Demilitarized Zones (DMZ). This sends received FIRs to a downstream component on a computer on the far side of a firewall.

### *Control component logctlsrv*

Component name	Component type	Description
logctlsrv	Control server	This component is used to control and configure all other CALA components.

Logctlsrv is controlled using the CLI (command line interface) **logctlcmd**. **Logctlcmd** reads configuration information from the file **logctlsrv.conf** and starts the process **logctlsrv**, which then takes control of the configuration management of all other CALA component.

## CLI *logctlcmd*

The Command Line Interface **logctlcmd** is used on all platforms supported by external control of the CALA component.

**NOTE** Starting and stopping the CALA component on Windows systems can be implemented by the Windows Service Manager (refer to CALA installation as an Windows service) as well.

## Supported *logctlcmd* commands

### startup

Starting the CALA component. A CALA Windows installation is started by the command **net start cala\_srv** if CALA is installed as a service.

### shutdown

Stopping the CALA component. A CALA Windows installation is stopped by the command **net stop cala\_srv** if CALA is installed as a service.

### restart

Restarts the CALA component

### status

Status query of the CALA component. In addition to the output of all status information for the installed CALA component, output includes information about the Tivoli environment employed as well as important CALA environment variables.

### reconfigure

Reconfiguration of the CALA component during runtime. Changes to the configuration (configuration file **logctl<sup>l</sup>srv.conf**) are taken into account at this time.

### maintenance\_on

Activation of the Maintenance Level (processing is delayed)

### maintenance\_off

Deactivation of the Maintenance Level (processing is restarted)

### test <logical\_name>

This command is employed in order to generate a CALA test event from a component. This test event can be used to test communication between the component (local on a computer or on a remote computer).

**NOTE** The CALA programs need a shared library which has to be available to them. See the following table for the name of the library and the environment variable to be set to its path.

operating system	filename of shared library	environment variable
MS Windows	libcala.dll	PATH
AIX	libcala.so	LIBPATH
Solaris	libcala.so	LD_LIBRARY_PATH
Linux	libcala.so	LD_LIBRARY_PATH
HP-UX	libcala.sl	SHLIB_PATH

## Generating test events

Apart from the emitters (senders), all components are able to generate test events. This means that communication can be tested between the individual components. Test events can be generated using the CLI call:

```
logctlcmd test <logical componentname>
```

## Possible component architecture (predecessors / successors)

The following table illustrates the possible component architecture with predecessors (previous stage) and successor (subsequent stage).

Componentname	predecessor	successor
ascfileread	-	tecfmtfilt, v2fmtfilt
ntevtlogread	-	tecfmtfilt v2fmtfilt
tecfmtfilt	ascfileread, ntevtlogread	any FIR processing component <sup>1</sup>
v2fmtfilt	ascfileread, ntevtlogread	any FIR processing component
snmpread	-	any FIR processing component
calamon	-	any FIR processing component
mssqlread	-	any FIR processing component
oracleread	-	any FIR processing component
jdbcread	-	any FIR processing component
javasrv /	-	any FIR processing component
msgclsfsrv	any FIR generating component <sup>2</sup>	any FIR processing component

<sup>1</sup> FIR processing components are:

- msgclsfsrv
- cmdemit
- reportemit
- tecfmitmit
- calaproxy
- snmpemit
- smtpemit
- jdbcemit
- remote component

<sup>2</sup> FIR generating components are:

- tecfmtfilt
- v2fmtfilt
- calamon
- snmpread
- mssqlread, oracleread, jdbcread



Componentname	predecessor	successor
cmdemit	any FIR generating component	-
reportemit	any FIR generating component	-
snmpemit	any FIR generating component	-
smtpeMIT	any FIR generating component	-
jdbcemIT	any FIR generating component	-
tecfmteMIT	any FIR generating component	tecifcsrv (end, sec, uns)
calaproxy	any FIR generating component	any FIR processing component
remote component	any FIR generating component	any FIR processing component, tecifcsrv (end, sec, uns)
tecifcsrv (end, sec, uns)	tecfmtfilt	-

- javasrv /
- msgclsfsrv
- calaproxy
- remote component

## Communication between CALA components

Communication between individual CALA components is based on TCP/IP communication with variable package size.

The data records read in (Logs, Windows Event Log, Syslog, etc.) are transferred by the filter processes to Filter Input Records (FIR) which form the basis for communication between all other CALA components.

When this standardized data object (FIR) is implemented for CALA component communication, CALA components are able to link up in almost any conceivable order.

The data (FIRs) can be transmitted through ports configured in any desired manner, and every component can also receive or transmit data via any desired number of ports.

## Default TCP ports used by CALA components

The following table shows the default TCP ports used by CALA components. Chapter 10 "Configuration file logctlsrv.conf" describes how the port settings can be changed.

component name	default port
logctlsrv	23861
logctlcmd	23860
ascfileread	23831
ntevtlogread	23832
calamon	23833
snmpread	23834
oracleread	23835
mssqlread	23836
jdbcread	23837
tecfmtfilt	23838
V2fmtfilt	23839
msgclsfsrv	23840
calaproxy	23841
tecfmtemit	23842
cmdemit	23843
reportemit	23844
snmpemit	23845
smtpemit	23846
tecifcsrv	23847
jdbcemit	23848

## Event caching

If a component loses contact with a downstream component during the transmission of events, these events are then stored in a cache file. In this case, the client process tries to reconnect to the server every 5 seconds.

As soon as a new connection can be established, the cached event can be transmitted. Once the transmission confirmation has been received, the cache entries are deleted.

Cache files are stored in the directory/folder defined by the environment variable `CALA_CACHE_DIR`. If `CALA_CACHE_DIR` has not been set, environment variables `TEMP` and `TMP` (with Windows also the `SystemRoot`) are evaluated. If none of these variables has been set, the cache file is stored in the current directory/folder.

**NOTE** The cache files are named `.<client>.<server>.cache`, so they may not be displayed by a normal `ls` call.

# CALA configuration

## Global information

This chapter provides detailed information about CALA configuration.

First, it describes global files like templates and general input files. These files are used for all datatypes in a configuration.

The following sub-chapters provide descriptions of all datatype definitions contained in this module.

### ***Who should read this chapter?***

We recommend this chapter for all who are interested in details about the logfiles processed by CALA.

### ***Related information***

For information about the directory structure on client see *Technical details of CALA configuration*.

## Global configuration files

### Configuration files

These files are located directly in the CALA installation directory (\$CALA\_DIR):

Name	Function
<code>logctlsrv.conf</code>	CALA configuration file; this file is generated from the input files and templates described in this chapter

### Templates

The template files serve as base for the main CALA configuration file `logctlsrv.conf`. The template is included in the package you select for installation.

Name	Function
<code>logctlsrv.client.templ_CLIENT</code>	template for FSM client configuration
<code>logctlsrv.complete.templ_SERVER</code>	template for FSM server configuration

### Configuration input files

These files are used in combination with the template files and the input files for the selected datatypes to create a client-specific configuration file. The input files are located in \$CALA\_DIR/**repos**.

Name	Function
<code>aux_fnislog.cala</code>	auxkey definitions for datatype <i>fnislog</i> . The auxkey definitions are used for dupe detection based on <i>error_id</i> and the starting characters of <i>original error text</i> (optional).
<code>remapper_fnislog.cala</code>	remapper definitions for datatype <i>fnislog</i> ; controls renaming of internal slots to the slots defined for the event console
<code>&lt;datatype&gt;.cala</code>	description files for each datatype; the datatypes available in this module are described in the following chapters

For a detailed description of the input files see *Technical details of CALA configuration*.

## Datatype `ibm_cm8_eventlog` - IBM CM8 Eventlog

Processing of log events created by IBM CM8. These events are read from the Library Server database.

### *Application and supported versions*

- IBM CM8 v8.3 and v8.4

### *Data sources*

#### *Databases*

- tables ICMSTSYSADMEVENTS and ICMSTITEMEVENTS from the CM8 database on the Library Server

### *How to activate CM8 logging*

Logging for IBM CM8 is always active.

### *Related files*

File	Function
<code>ibm_cm8_eventlog.cala</code>	Input file for configurator
<code>ibm_cm8_eventlog_class.map</code>	classmap file; maps CM8 EVENTCODE to classnames
<code>ibm_cm8_eventlog_filter_↓ snmp.map</code>	set severity in slot spectype or value2 depending on SNMP version (non-Tivoli based version only)
<code>ibm_cm8_eventlog_sev.map</code>	set severity and create entry for slot msg
<code>ibm_cm8_eventlog_sntp.map</code>	create slots required for smtpemit (non-Tivoli based version only)
<code>ibm_cm8_eventlog_sntp.rmp</code>	no email for HARMLESS and WARNING events (non-Tivoli based version only)
<code>ibm_cm8_eventlog_status.map</code>	set severity in slot status; used for jdbcemit (non-Tivoli based version only)

## Additional information

### Event filtering

- events can be filtered by `EVENTCODE`
- filtering of events can be activated in the **Set configuration variables...** dialog in the FSM client installer
- in the section **IBM Content Manager Version 8 Eventlog**, separate filters can be specified for events from the table `ICMSTSYSADMEVENTS` and `ICMSTITEMEVENTS`

### Adjusting severities

- the default class for all events is `IBM_CM8_EVENTLOG_Error`
- to change the severity for certain events, change the file `ibm_cm8_eventlog_class.map`; each line in this file contains an eventcode from the database and the value for `$CLASS` that must be associated with the eventcode  
Sample entry:  
`202 IBM_CM8_EVENTLOG_Info`
- if other values for `$CLASS` are used than the default classes, the file `ibm_cm8_eventlog_sev.map` must be adjusted as well as this file maps the value of `$CLASS` to the severity that is displayed in the Webconsole  
Default classes:  
`IBM_CM8_EVENTLOG_Info`  
`IBM_CM8_EVENTLOG_Warning`  
`IBM_CM8_EVENTLOG_Error`  
`IBM_CM8_EVENTLOG_Fatal`



## Datatype `ibm_cmcs_archive` - IBM CS archive log

Processing of IBM Common Store archive logfiles.

### *Application and supported versions*

- Common Store 8.3

### *Data sources*

Logfiles

- `csserror.log`  
The error log file is created in the directory that the `INSTANCEPATH` keyword points to. The keyword is set in the server configuration file; usually `archint.ini`.
- The logfile for the archiving task can be analyzed. Its name is defined in the archive task script.
- The logfile for the retrieving task can be analyzed. Its name is defined in the retrieve task script.

### *How to activate IBM CS archive log*

### *Related files*

File	Function
<code>ibm_cmcs_archive.cala</code>	Input file for configurator
<code>ibm_cmcs_archive.v2s</code>	Formatfile
<code>ibm_cmcs_archive_dup.map</code>	dupe detection based on error_id
<code>ibm_cmcs_archive_filter_↓ snmp.map</code>	set severity in slot spectype or value2 depending on snmp version (non-Tivoli based version only)
<code>ibm_cmcs_archive_status.map</code>	set severity in slot status; used for mysqlmit (non-Tivoli based version only)
<code>ibm_cmcs_archive_sntp.map</code>	create slots required for smtpmit (non-Tivoli based version only)
<code>ibm_cmcs_archive_sntp.rmp</code>	no email for HARMLESS and WARNING events (non-Tivoli based version only)

## ***Additional information***

## Datatype `ibm_cmcs_error` - IBM CM8 agent and CS server error log

Processing of event log files created by IBM Content Manager Version 8 agent and Common Store server.

### *Application and supported versions*

- Content Manager 8.3
- Common Store 8.3

### *Data sources*

Logfiles

- `<INSTANCEPATH>/cm8errors.log`
- `csserror.log`; the path for the logfile can be specified in `archint.ini`

### *How to activate IBM CM8 and CS logfiles*

### *Related files*

File	Function
<code>ibm_cmcs_error.cala</code>	Input file for configurator
<code>ibm_cmcs_error.v2s</code>	Formatfile
<code>ibm_cmcs_error_dup.map</code>	dupe detection based on error_id
<code>ibm_cmcs_error_filter_snmp.map</code>	set severity in slot spectype or value2 depending on snmp version (non-Tivoli based version only)
<code>ibm_cmcs_error_status.map</code>	set severity in slot status; used for mysqlmit (non-Tivoli based version only)
<code>ibm_cmcs_error_smtmp.map</code>	create slots required for smtpemit (non-Tivoli based version only)
<code>ibm_cmcs_error_smtmp.rmp</code>	no email for HARMLESS and WARNING events (non-Tivoli based version only)

## ***Additional information***

## Datatype `ibm_cmcs_retrieve` - IBM CS retrieve log

Processing of IBM Common Store retrieve logfiles.

### *Application and supported versions*

- Common Store 8.3

### *Data sources*

Logfiles

- 

### *How to activate IBM CS retrieve log*

### *Related files*

File	Function
<code>ibm_cmcs_retrieve.cala</code>	Input file for configurator
<code>ibm_cmcs_retrieve.v2s</code>	Formatfile
<code>ibm_cmcs_retrieve_dup.map</code>	dupe detection based on error_id
<code>ibm_cmcs_retrieve_filter_↓ snmp.map</code>	set severity in slot spectype or value2 depending on snmp version (non-Tivoli based version only)
<code>ibm_cmcs_retrieve_status_↓ map</code>	set severity in slot status; used for mysqlmit (non-Tivoli based version only)
<code>ibm_cmcs_retrieve_smtmp.map</code>	create slots required for smtpermit (non-Tivoli based version only)
<code>ibm_cmcs_retrieve_smtmp.rmp</code>	no email for HARMLESS and WARNING events (non-Tivoli based version only)

### *Additional information*

## Datatype `ibm_cmod_log` - IBM CM On Demand Logfile

Processing of log events created by IBM CM On Demand. These events are read from the database.

### *Application and supported versions*

- IBM CM On demand 8.3, 8.4

### *Data sources*

#### *Databases*

- table ODADMIN.SL2 from the database ARCHIVE1 for logging

### *How to activate CM On Demand logging*

Logging for IBM CM On demand is always active.

### *Related files*

File	Function
<code>ibm_cmod_log.cala</code>	Input file for configurator
<code>ibm_cmod_log_class.map</code>	classmap file; maps CM On Demand SEVERITY to class-names
<code>ibm_cmod_log_dup.map</code>	duplicate detection
<code>ibm_cmod_log_filter_snmp.map</code>	set severity in slot spectype or value2 depending on SNMP version (non-Tivoli based version only)
<code>ibm_cmod_log_sev.map</code>	set severity and create entry for slot msg
<code>ibm_cmod_log_sntp.map</code>	create slots required for sntpemit (non-Tivoli based version only)
<code>ibm_cmod_log_sntp.rmp</code>	no email for HARMLESS and WARNING events (non-Tivoli based version only)
<code>ibm_cmod_log_status.map</code>	set severity in slot status; used for jdbcemit (non-Tivoli based version only)

## Datatype `ibm_cmrm_asyncr` - IBM CM RM Asyncr logfiles

Processing of IBM Content Manager Resource Manager Asyncr logfiles.

### *Application and supported versions*

- Content Manager 8.3

### *Data sources*

Logfiles

- e.g. `/ddata/cm_logs/icmrm_asyncr.logfile`; path and filename are configurable

### *How to activate IBM CM RM Asyncr logfiles*

### *Related files*

File	Function
<code>ibm_cmrm_asyncr.cala</code>	Input file for configurator
<code>ibm_cmrm_asyncr.v2s</code>	Formatfile
<code>ibm_cmrm_asyncr_dup.map</code>	dupe detection based on error_id
<code>ibm_cmrm_asyncr_filter_snmp.map</code>	set severity in slot spectype or value2 depending on snmp version (non-Tivoli based version only)
<code>ibm_cmrm_asyncr_status.map</code>	set severity in slot status; used for mysqlmit (non-Tivoli based version only)
<code>ibm_cmrm_asyncr_smtp.map</code>	create slots required for smtpemit (non-Tivoli based version only)
<code>ibm_cmrm_asyncr_smtp.rmp</code>	no email for HARMLESS and WARNING events (non-Tivoli based version only)

### *Additional information*

## Datatype `ibm_cmr_migrator` - IBM CM RM Migrator Logfiles

Processing of IBM Content Manager Resource Manager Migrator logfiles.

### *Application and supported versions*

- Content Manager 8.3

### *Data sources*

Logfiles

- e.g. `/ddata/cm_logs/migrator.logfile`; path and filename are configurable

### *How to activate IBM CM RM Migrator logfiles*

### *Related files*

File	Function
<code>ibm_cmr_migrator.cala</code>	Input file for configurator
<code>ibm_cmr_migrator.v2s</code>	Formatfile
<code>ibm_cmr_migrator_dup.map</code>	dupe detection based on error_id
<code>ibm_cmr_migrator_filter_↓ snmp.map</code>	set severity in slot spectype or value2 depending on snmp version (non-Tivoli based version only)
<code>ibm_cmr_migrator_status_↓ map</code>	set severity in slot status; used for mysqlmit (non-Tivoli based version only)
<code>ibm_cmr_migrator_smtp.map</code>	create slots required for smtpemit (non-Tivoli based version only)
<code>ibm_cmr_migrator_smtp.rmp</code>	no email for HARMLESS and WARNING events (non-Tivoli based version only)

### *Additional information*



## Customization Issues

This chapter contains the most common customization tasks for CALA. For a complete description of all CALA options see *CALA components*.

### Globalization settings (NLS support)

In general log file and Eventlog entries are processed with the character settings of the system. The globalization support additionally allows processing logfiles containing characters that are not identical with the system settings.

Example: To process a logfile containing character set 'ISO-2022-JP' characters the following settings need to be used:

- `C:/TEMP/logs/mylogfile.log:ISO-2022-JP`

The complete list of all supported character sets are added as an appendix.

### Adding logfile definitions

The logfiles for each datatype are defined in the corresponding .cala file. The datatype controls which format file will be used to parse the logfile. If you want to add logfiles to the CALA configuration, you must make sure that they can be processed with the format files associated with the datatype.

These are the default logfile settings:

Datatype *fnislog* - FileNet ELOG Logfiles and HPII / MRII Logfiles

- `${__FNIS__FN_LOC_PATH}/logs/elogs/elogYYYYMMDD`
- `${__FNIS__FN_LOC_PATH}/logs/elogs/elYYYYMMDD`
- `${__FNIS__HPII_PATH}/journals/*YYYYMMDD`

Datatype *fdw4log* - FileNet DocWarehouse 4.0 Logfiles

- `/tmp/srvlink.log`
- `${__FNIS__SRVLINK_PATH}/srvlink.log`
- `${SystemRoot}/srvlink.log`

To add more logfiles for a datatype, unpack the corresponding .tar.gz archive, edit the .cala file directly and recreate the .tar.gz archive.

### Adding message map definitions

You can add your own message map definitions to the CALA configuration.

First, create a message map file according to the map files provided with the modules. The name of the message map file must follow the naming convention `<datatype>_<suffix>.map`.

Now add the new map entry to the appropriate .cala file.

You must edit the file `<datatype>.cala` which is located in the directory `temp` in the configuration tar.gz archive (see *Technical details of CALA configuration*, section *Directory structure in FSM installation archives* for details of the configuration archives).

Add a block for the message map file. The block must look like this:

```
MAP_NAME_<n>=<suffix>
MAP_PRI_TYPE_<n>=v2
MAP_TARGET_<n>=<target>
```

`<n>` is the number of the map. Note that the sequence of the `MAP_` entries in the .cala file is relevant for the processing of the message maps. Make sure that the numbering of the map entries is unique.

The `MAP_TARGET` entry can be copied from the existing map entries.

## Adjusting port numbers

The components of the FSM CALA use ports for their communication. If you want to change the standard ports, make sure to change the `-P <portnum>` entry as well as the corresponding `port!<portnum>` specification. You can change the port numbers after configuration / distribution of CALA if you want to change them for one target only. If you want to change the port numbers for all targets, you can edit the template file.

Component	Standard port number
ascfileread	23831
ntevtlogread	23832
tecfmfiltr	23838
v2fmfiltr	23839
mssqlread	23836
oracleread	23835
calamon	23833
javasrv/pchread	23849
snmpread	23834
msgclsfsrv	23840
tecfmtemit	23842
tecifcsrv	23847
snmpemit	23845
smtpemit	23846
cmdemit	23843
reportemit	23844

If your CALA installation is client-server based, the client component uses the port specified for `msgclsfsrv` in the server template as remote port number (23840 in the table above).

## Activate logging for CALA

You can turn on logging for each of the CALA components. Add the `-d:<loglevel>:<logfilename>` parameter to the configuration entry of the component.

The `<loglevel>` can be a number between 0 and 9. The lower the loglevel is, the more will be written to the logfile.

Note that the logfile may grow very fast if you choose a high loglevel and process large logfiles.

The following configuration fragment shows an entry for the component `ascfileread` with activated logging. Note that the referenced subdirectory `logs` must already exist to allow creation of the logfile `ascfileread.log`.

```
ascfileread=run!ascfileread -P 11001 -d:3:logs/ascfileread.log,port!11001,targetsv2fmtfilt,↓  
pathlist!1;/fnsw/local/logs/elogs,ptrnlist!1;elogYYYYMMDD,assoc!1;1;v2;fnislog,conf!port;↓  
run;targets;pathlist;ptrnlist;assoc
```

## Activate heartbeat events for CALA

CALA heartbeat is enabled by default. The events are handled automatically by the Webconsole.

The heartbeat is controlled by the `-ZHEARTBEAT_PERIOD=<secs>` parameter in the configuration entry of the component that creates events of the class `CALA_HEARTBEAT_OK`.

`<secs>` is the interval between two heartbeat events. If it is set to zero, no heartbeat events will be generated.

The following configuration fragment shows an entry for the component `v2fmtfilt` with activated heartbeat. The heartbeat interval is set to 300 seconds (5 minutes).

```
v2fmtfilt=run!v2fmtfilt -P 11004 -ZHEARTBEAT_PERIOD=300,port!11004,targetsv2fmtfilt,↓  
formatlist!fnislog;fmt/fnislog.v2s,conf!port;run;targets;formatlist
```

This option is appropriate for the filter component `v2fmtfilt` in a complete or a client configuration and for the server component `msgclsfsrv` in a server configuration.

## Adding environment settings for CALA

If you need to set client specific environment variables for CALA, you can create an additional environment file:

UNIX:	\$CALA_DIR/cala_env.sh
Windows:	%CALA_DIR%\cala_env.cmd

No additional changes are required. The new file will be used as soon as the startup script is executed the next time (UNIX) or the cala\_srv service is restarted (Windows).

## Sample configuration

This is a sample configuration files that processes ELOG logfiles on a UNIX system. The configuration is complete, containing the client (reader) component as well as the server (message classification and TEC interface) components.

```
#operating-system: sun-solaris
#name of configuration: Generated by CALACFG
#user:
#password:
serverlist=ascfileread,v2fmtfilt,msgclsfsrv,tecfmtemit,tecifcsrv
ascfileread=run!ascfileread -P 11001,port!11001,target!v2fmtfilt,pathlist!1;/fnsw/local/log.
s/elogs,ptrnlist!1;elogYYYYMMDD,assoc!1;1;v2;fnislog,conf!port;run;target!v2fmtfilt,pathlist;
ptrnlist;assoc
v2fmtfilt=run!v2fmtfilt -P 11004,port!11004,target!msgclsfsrv,formatlist!fnislog;
fmt/fnislog.v2s,conf!port;run;target!msgclsfsrv,formatlist
msgclsfsrv=run!msgclsfsrv -P 11009,port!11009,target!tecfmtemit,type!fnislog_v2_mct,
auxkeys!aux_fnislog_0_8;aux_fnislog_0_15;aux_fnislog_0_6;aux_fnislog_0_2;aux_fnislog_0_
0;aux_fnislog_0_20;aux_fnislog_generell,remappers!tecfmtemit_remap,conf!port;run;
target!tecfmtemit_remap,conf!port;run;
target!tecfmtemit_remap,conf!port;run;
target!tecfmtemit_remap,conf!port;run;
fnislog_v2_mct=type!v2;fnislog,handledby!tecfmtemit,msgmaps!misc/fnislog_evt.map;fnislog_
evt_map;misc/fnislog_dup.map;fnislog_dup_map;misc/fnislog_except_0_15.map;fnislog_except_0_
15_map;misc/fnislog_except_0_8.map;fnislog_except_0_8_map;misc/fnislog_except_0_4.
map;fnislog_except_0_4_map;misc/fnislog_filter.map;fnislog_filter_map
fnislog_evt_map=key!error_id;L,fields!severity;msg;error_cause;corrective_action
fnislog_dup_map=key!error_id;L,fields!$DUPEKEY;$ESCAT;$ESCLEV;$ESCCNT
fnislog_except_0_15_map=key!error_id;L;original_error_text;F0T15w16,fields!severity;
msg;error_cause;corrective_action
fnislog_except_0_8_map=key!error_id;L;original_error_text;F0T8w9,fields!severity;msg;error_
cause;corrective_action
fnislog_except_0_4_map=key!error_id;L;original_error_text;F0T4w5,fields!severity;msg;error_
cause;corrective_action
fnislog_filter_map=key!severity;L,fields!$CLASS;severity
# auxkey definitions
aux_fnislog_0_0=error_id;L
aux_fnislog_0_2=error_id;L;original_error_text;F0T2
aux_fnislog_0_20=error_id;L;original_error_text;F0T20
aux_fnislog_0_6=error_id;L;original_error_text;F0T6
aux_fnislog_0_4=error_id;L;original_error_text;F0T4
aux_fnislog_0_15=error_id;L;original_error_text;F0T15
aux_fnislog_0_8=error_id;L;original_error_text;F0T8
aux_fnislog_generell=error_id;L;original_error_text;L
# remapper definitions
tecfmtemit_remap=for!tecfmtemit,fieldalias!$ESCCNT;occurrences_before_send
tecfmtemit=run!tecfmtemit -P 11010,port!11010,target!tecifcsrv,conf!port;run;target!
tecifcsrv=run!tecifcsrvend -P 11011 -h @EventServer ,port!11011,conf!port;run
```

## Appendix A. Local Environment File fsm4ibm\_srv\_env.sh

This table provides an overview for the information stored in the local environment file `fsm4ibm_srv_env.sh` that is created on each server that hosts IBM IM products.

**NOTE** Note: These variables cannot be used in the logfile definitions for the FSM CALA (CALA) yet.

These are the global settings:

Variable name	Description
FSM4IBM_SYS_n	system name
FSM4IBM_SYS_n_SRVLIST	list of servers for this system
FSM4IBM_SYS_n_WEBENVLIST	list of web environments
FSM4IBM_SYS_n_IICELIST	list of configured IICE systems
FSM4IBM_SYS_n_CMLIST	list of configured Content Manager systems
FSM4IBM_SYS_n_CSLIST	list of configured Common Store systems
FSM4IBM_SYS_n_CMODLIST	list of configured Content Manager On Demand systems
FSM4IBM_SYSLIST	list of systems on this server

These are the settings for server instances:

Variable name	Description
FSM4IBM_SYS_n_SRV_n	server name for remote servers that are monitored by this host; <i>localhost</i> if this server monitors itself
FSM4IBM_SYS_n_SRV_n_IS_LOCALHOST	<i>NO</i> for remote servers, <i>YES</i> for localhost
FSM4IBM_SYS_n_SRV_n_DESCRIPTION	server description
FSM4IBM_SYS_n_SRV_n_INST_	instance name
FSM4IBM_SYS_n_SRV_n_INST_n_JMX_APPSRV_↓ TYPE	application server type
FSM4IBM_SYS_n_SRV_n_INST_n_JMX_SERVICE_↓ URL	JMX service URL; specific for the application server
FSM4IBM_SYS_n_SRV_n_INST_n_JMX_PORT	JMX connection port
FSM4IBM_SYS_n_SRV_n_INST_n_JMX_USER	User to use for the JMX connection

Variable name	Description
FSM4IBM_SYS_n_SRV_n_INST_n_JMX_PASSWD	Encrypted password of JMX_USER
FSM4IBM_SYS_n_SRV_n_INST_n_JMX_TMOUT	Timeout for connection
FSM4IBM_SYS_n_SRV_n_INST_n_JMX_JAVAPATH	Application server specific Java path
FSM4IBM_SYS_n_SRV_n_INST_n_JMX_LIBPATH	Application server specific library path
FSM4IBM_SYS_n_SRV_n_INST_n_JMX_CONDATA	Application server specific connection data
FSM4IBM_SYS_n_SRV_n_INST_n_JMX_↓ ADDITIONAL_OPTIONS	For all other JMX specific settings
FSM4IBM_SYS_n_SRV_n_INSTLIST	list of instances

These are the settings for Webenvironments:

Variable name	Description
FSM4IBM_SYS_n_WEBENV_n	Web environment name
FSM4IBM_SYS_n_WEBENV_n_ADMIN_SERVER	The administrative server of the web environment
FSM4IBM_SYS_n_WEBENV_n_INSTLIST	Instances of the web environment. The instances are listed with their server environment specific keys

These are the settings for IICE:

Variable name	Description
FSM4IBM_SYS_n_IICE_n	IICE system name
FSM4IBM_SYS_n_IICE_n_SRV_n	server name
FSM4IBM_SYS_n_IICE_n_SRV_n_DATA_RMI_↓ HOSTNAME	Hostname of the II CE server for RMI connects
FSM4IBM_SYS_n_IICE_n_SRV_n_INST_n	server instance name
FSM4IBM_SYS_n_IICE_n_SRV_n_INST_n_DATA_↓ INSTPATH	Installation path of the II CE instance at a specific server
FSM4IBM_SYS_n_IICE_n_SRV_n_INST_n_DATA_↓ RMIBRIDGE_SCRIPT	The RMI bridge start script of the II CE instance at a specific RMI Proxy Connector server ( <i>FSM4IBM_SYS_n_IICE_n_SRV_n</i> ). Not used for the II CE server running in an application server. The script must be located in <i>FSM4IBM_SYS_n_IICE_n_SRV_n_↓INST_n_DATA_INSTPATH/bin/</i> .

Variable name	Description
FSM4IBM_SYS_n_IICE_n_SRV_n_INST_n_DATA_↓ RMI_PORT	RMI port for RMI connects to an RMI instance at a specific server ( <i>FSM4IBM_SYS_n_IICE_n_SRV_n</i> ).
FSM4IBM_SYS_n_IICE_n_SRV_n_INST_n_DATA_↓ RMI_JAVAPATH	Path to a Java installation, that is used for RMI connects to an RMI instance at a specific server ( <i>FSM4IBM_XX_SRV</i> ).
FSM4IBM_SYS_n_IICE_n_SRV_n_INST_n_DATA_↓ RMI_LOGFILE	Optional path to an II CE logfile of an instance at a specific server ( <i>FSM4IBM_SYS_n_IICE_n_SRV_n</i> ).
FSM4IBM_SYS_n_IICE_n_SRV_n_INSTLIST	list of server instances
FSM4IBM_SYS_n_IICE_n_SRVLIST	list of servers
FSM4IBM_SYS_n_IICE_n_WEBENV_n_INST_n	Web-environment instance name
FSM4IBM_SYS_n_IICE_n_WEBENV_n_INST_n_↓ JMX_APPLICATION_NAME	Application server application name used for JMX monitors.
FSM4IBM_SYS_n_IICE_n_WEBENV_n_INST_n_↓ JMX_WARFILE_NAME	Application WAR file name.
FSM4IBM_SYS_n_IICE_n_WEBENV_n	Web-environment of an II CE application server installation.
FSM4IBM_SYS_n_IICE_n_WEBENV_n_INSTLIST	list of webenv instances
FSM4IBM_SYS_n_IICE_n_WEBENVLIST	list of web environments

These are the settings for Content Manager:

Variable name	Description
FSM4IBM_SYS_n_CM_n	CM system name
FSM4IBM_SYS_n_CM_n_SRV_n	server name
FSM4IBM_SYS_n_CM_n_SRV_n_DATA_IBMCMROOT	Root directory of a CM installation; equal to the environment variable IBMCMROOT.
FSM4IBM_SYS_n_CM_n_SRV_n_LIB_SRV_DATA_↓ DBTYPE	Type of the CM library server database.
FSM4IBM_SYS_n_CM_n_SRV_n_LIB_SRV_DATA_↓ DBPATH	Path to the root directory of the CM library server database installation.
FSM4IBM_SYS_n_CM_n_SRV_n_LIB_SRV_DATA_↓ DBNAME	CM library server database name.
FSM4IBM_SYS_n_CM_n_SRV_n_LIB_SRV_DATA_↓ DBNAMEREMOTE	CM library server database name for remote access.
FSM4IBM_SYS_n_CM_n_SRV_n_LIB_SRV_DATA_↓ DBUSEROS	CM library server's operating system user for database access.
FSM4IBM_SYS_n_CM_n_SRV_n_LIB_SRV_DATA_↓	CM library server admin user account.



Variable name	Description
DBUSERMAINT	
FSM4IBM_SYS_n_CM_n_SRV_n_LIB_SRV_DATA_↓ DBPASSWDMAINT	Password of the admin user account specified in <i>FSM4IBM_SYS_n_CM_n_SRV_n_LIB_↓SRV_DATA_DBUSERMAINT</i>
FSM4IBM_SYS_n_CM_n_SRV_n_LIB_SRV_DATA_↓ DBUSERRUNTIME	CM library server database runtime user account.
FSM4IBM_SYS_n_CM_n_SRV_n_LIB_SRV_DATA_↓ DBTWOTASK	TWO_TASK variable for remote Oracle
FSM4IBM_SYS_n_CM_n_SRV_n_RM_SRV_DATA_↓ DBTYPE	Type of the CM resource manager database.
FSM4IBM_SYS_n_CM_n_SRV_n_RM_SRV_DATA_↓ DBPATH	Path to the root directory of the CM resource manager database installation
FSM4IBM_SYS_n_CM_n_SRV_n_RM_SRV_DATA_↓ DBNAME	CM resource manager database name.
FSM4IBM_SYS_n_CM_n_SRV_n_RM_SRV_DATA_↓ DBNAMEREMOTE	CM resource manager database name for remote access.
FSM4IBM_SYS_n_CM_n_SRV_n_RM_SRV_DATA_↓ DBUSEROS	CM resource manager's operating system user for database access.
FSM4IBM_SYS_n_CM_n_SRV_n_RM_SRV_DATA_↓ DBUSERMAINT	CM resource manager admin user account.
FSM4IBM_SYS_n_CM_n_SRV_n_RM_SRV_DATA_↓ DBPASSWDMAINT	Password of the admin user account specified in <i>FSM4IBM_SYS_n_CM_n_SRV_n_RM_↓SRV_DATA_DBUSERMAINT</i> .
FSM4IBM_SYS_n_CM_n_SRV_n_RM_SRV_DATA_↓ DBUSERRUNTIME	CM resource manager database runtime user account.
FSM4IBM_SYS_n_CM_n_SRV_n_RM_SRV_DATA_↓ DBTWOTASK	TWO_TASK variable for remote Oracle
FSM4IBM_SYS_n_CM_n_SRVLIST	list of servers
FSM4IBM_SYS_n_CM_n_WEBENV_n_INST_n	Web-environment instance name.
FSM4IBM_SYS_n_CM_n_WEBENV_n_INST_n_RM_↓ JMX_APPLICATION_NAME	Application server application name used for JMX monitors. Only valid for the Ressource Manager.
FSM4IBM_SYS_n_CM_n_WEBENV_n_INST_n_RM_↓ JMX_WARFILE_NAME	Application WAR file name. Only valid for the Ressource Manager.
FSM4IBM_SYS_n_CM_n_WEBENV_n	Web-environment of a CM resource manager application server installation.
FSM4IBM_SYS_n_CM_n_WEBENV_n_INSTLIST	list of webenv instances
FSM4IBM_SYS_n_CM_n_WEBENVLIST	list of web environments

These are the settings for Common Store:

Variable name	Description
FSM4IBM_SYS_n_CS_n	CS system name
FSM4IBM_SYS_n_CS_n_SRV_n	server name
FSM4IBM_SYS_n_CS_n_SRV_n_CSTYPE	CS type ( <i>CSLD</i> or <i>CSX</i> ).
FSM4IBM_SYS_n_CS_n_SRV_n_INST_n	CS instance at a specific server ( <i>FSM4IBM_SYS_n_CS_n_SRV_n</i> ).
FSM4IBM_SYS_n_CS_n_SRV_n_INST_n_↓ ARCHINTINIT_PATH	Directory of the archint.ini file of a specific CS instance ( <i>FSM4IBM_SYS_n_CS_n_SRV_n_INST_n</i> ) at a specific server ( <i>FSM4IBM_SYS_n_CS_n_SRV_n</i> ).
FSM4IBM_SYS_n_CS_n_SRV_n_INST_n_BINPATH	Path to the CS binaries of a specific CS instance ( <i>FSM4IBM_CS_SRV_INST</i> ) at a spe- cific server ( <i>FSM4IBM_SYS_n_CS_n_SRV_n</i> ).
FSM4IBM_SYS_n_CS_n_SRV_n_INST_n_ARCHIVE_↓ TASK_PATH	Path to the archive task script of a specific CS instance ( <i>FSM4IBM_SYS_n_CS_n_SRV_n_INST_n</i> ) at a specific server ( <i>FSM4IBM_SYS_n_CS_n_SRV_n</i> ).
FSM4IBM_SYS_n_CS_n_SRV_n_INST_n_↓ RETRIEVE_TASK_PATH	Path to the retrieve task script of a specific CS instance ( <i>FSM4IBM_SYS_n_CS_n_SRV_n_INST_n</i> ) at a specific server ( <i>FSM4IBM_SYS_n_CS_n_SRV_n</i> ).
FSM4IBM_SYS_n_CS_n_SRV_n_INSTLIST	list of instances
FSM4IBM_SYS_n_CS_n_SRVLIST	list of servers

These are the settings for CMOD:

Variable name	Description
FSM4IBM_SYS_n_CMOD_n	CMOD system name
FSM4IBM_SYS_n_CMOD_n_SRV_n	server name
FSM4IBM_SYS_n_CMOD_n_SRV_n_DATA_↓ INSTPATH	Installation path of a CMOD for a specific server ( <i>FSM4IBM_SYS_n_CMOD_n_SRV_n</i> ).
FSM4IBM_SYS_n_CMOD_n_SRV_n_DATA_USEROS	Operating user for CMOD.
FSM4IBM_SYS_n_CMOD_n_SRV_n_DATA_↓ PASSWDUSEROS	Password for the operating user for CMOD ( <i>FSM4IBM_SYS_n_CMOD_n_SRV_n_↓ DATA_USEROS</i> ).
FSM4IBM_SYS_n_CMOD_n_SRV_n_DATA_DBTYPE	Type of the CMOD database.
FSM4IBM_SYS_n_CMOD_n_SRV_n_DATA_DBPATH	Path to the root directory of the CMOD database installation.
FSM4IBM_SYS_n_CMOD_n_SRV_n_DATA_DBNAME	CMOD database name.
FSM4IBM_SYS_n_CMOD_n_SRV_n_DATA_↓	CMOD database name for remote access.

Variable name	Description
DBNAMEREMOTE	
FSM4IBM_SYS_n_CMODO_n_SRV_n_DATA_↓ DBUSEROS	CMOD 's operating system user for database access.
FSM4IBM_SYS_n_CMODO_n_SRV_n_DATA_↓ DBUSERMAINT	CMOD admin user account.
FSM4IBM_SYS_n_CMODO_n_SRV_n_DATA_↓ DBPASSWDMAINT	Password of the admin user account specified in <i>FSM4IBM_SYS_n_CMODO_n_SRV_n_↓DATA_↓DBUSERMAINT</i> .
FSM4IBM_SYS_n_CMODO_n_SRV_n_DATA_↓ DBUSERRUNTIME	CMOD database runtime user account.
FSM4IBM_SYS_n_CMODO_n_SRV_n_DATA_↓ DBTWOTASK	TWO_TASK variable for remote Oracle
FSM4IBM_SYS_n_CMODO_n_SRV_n_DATA_TSM_↓ DEFINED	<i>true</i> if TSM is configured; otherwise <i>false</i>
FSM4IBM_SYS_n_CMODO_n_SRV_n_DATA_TSM_↓ INSTPATH	TSM installation path; optional, can be empty.
FSM4IBM_SYS_n_CMODO_n_SRV_n_DATA_↓ JAVAPATH	Path to a JRE installation.
FSM4IBM_SYS_n_CMODO_n_SRV_n_DATA_↓ JARFILES	JAR files needed to access the CMOD (e.g. the JDBC driver's jar file).
FSM4IBM_SYS_n_CMODO_n_SRV_n_DATA_JDBCURL	JDBC URL used for the database connect.
FSM4IBM_SYS_n_CMODO_n_SRVLIST	list of servers
FSM4IBM_SYS_n_CMODO_n_WEBENV_n_INST_n	Web-environment instance name.
FSM4IBM_SYS_n_CMODO_n_WEBENV_n_INST_n_↓ JMX_APPLICATION_NAME	Application server application name used for JMX monitors.
FSM4IBM_SYS_n_CMODO_n_WEBENV_n_INST_n_↓ JMX_WARFILE_NAME	Application WAR file name.
FSM4IBM_SYS_n_CMODO_n_WEBENV_n	Web-environment of the deployed CMOD parts.
FSM4IBM_SYS_n_CMODO_n_WEBENV_n_INSTLIST	list of webenv instances
FSM4IBM_SYS_n_CMODO_n_WEBENVLIST	list of web environments

## Appendix B. Copyright notice

### IBM FileNet System Monitor (18 March 2011)

Copyright © 2000-2011 CENIT AG, Germany; Copyright © 2005, 2011 IBM Corp., USA including this documentation and all software.

No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without prior written permission of the copyright owners. The copyright owners grants you limited permission to make hardcopy or other reproductions of any machine-readable documentation for your own use, provided that each such reproduction shall carry the original copyright notice. No other rights under copyright are granted without prior written permission of the copyright owners. The document is not intended for production and is furnished as is without warranty of any kind. *All warranties on this document are hereby disclaimed including the warranties of merchantability and fitness for a particular purpose.*

Note to U.S. Government Users Documentation related to restricted rights Use, duplication or disclosure is subject to restrictions set forth in GSA

---

## Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106-0032, Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
J46A/G4  
555 Bailey Avenue  
San Jose, CA 95141-1003  
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

---

## Trademarks

IBM, the IBM logo, and [ibm.com](http://ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.