Securing Caché, Ensemble, And Operating System Resources

Version 2018.1
2020-11-13

InterSystems Corporation 1 Memorial Drive Cambridge MA 02142 www.intersystems.com

Securing Caché, Ensemble, And Operating System Resources
Caché Version 2018.1 2020-11-13

For Support questions about any InterSystems products, contact:

InterSystems Worldwide Response Center (WRC)
Tel: +1-617-621-0700
Tel: +44 (0) 844 854 2917
Email: support@InterSystems.com

Table of Contents

Securing Caché, Ensemble, And Operating System Resources

1 Introduction

This document is intended to assist customers who need to harden the operating system on which a Caché or Ensemble instance runs to reduce the potential attack surface available to an intruder. The operating system services required by a Caché instance are described. An inventory of various types of Caché process is provided and their purpose is explained. Methods for identifying the function of Caché processes in a running instance are described. Instructions are provided for removing or disabling optional Caché processes that not required. External processes, running programs other than `cache` on UNIX® or `cache.exe` on Windows, required by a running Caché instance are described. TCP and UDP ports used by Caché internal and external processes are listed and their purpose described.

2 Caché Processes

Most processes comprising a Caché instance will be executing the cache executable on UNIX® and the cache.exe executable on Windows which resides in the `bin` directory under the installation directory. A running instance uses a number of system processes to coordinate and support the processes running customer code. Caché processes can be examined using the Management Portal > System Operation > Processes.

2.1 Core Processes

Core system processes are started early in the initialization of an instance and have no value in the `User` column. These processes can be identified by the value in the `Routine` column which, in the case of system processes, does not always contain the name of a Caché routine. The following core system processes are listed by the name in the `Routine` column.

- CONTROL – Creates and initializes shared memory and provides various control functions.
- WRTDMN – The write daemon performs all writes to databases and WIJ.
- GARCOL – Garbage collects large kills.
- JRNDMN – Performs journal writes.
- EXPDMN – Performs database expansions.
- SWRTDMN – Write daemon secondary workers.
- MONITOR – Writes alerts to alert file and transmits email alerts.
- CLNDMN – Detects dead processes and cleans up stranded resources.
- RECEIVE – Manages ECP worker processes.
- ECPWork – ECP worker process.
- %SYS.SERVER – SuperServer process which accepts TCP requests and dispatches workers to serve them.
- %CSP.Daemon – Manages expiration of CSP sessions.

- LMFMON – Monitors the Caché license and sends usage data to the license server over UDP.
- %SYS.Monitor.xxx – Various system monitor workers.
- SYS.Monitor.xxx – Writes alerts to alert file and transmits email alerts (Version 2015.2).

The core system processes should not be stopped. Doing so will disrupt the normal operation of a Caché instance. It is possible, if ECP is not used, to prevent the ECPWork process from being started by setting appropriate values in the configuration file. From the management portal, select System Administration > Configuration > Connectivity > ECP settings and set the maximum number of application and data servers to zero. Then disable the ECP service.

A number of other Caché system processes are started after the core system processes. Many are started dynamically. These processes have a value displayed in the User column. Many of them are optional and are not started unless needed or configured. These processes can usually be identified by the values of the `Routine`, `User`, and `Client EXE` columns of the process display.

The task manager process (TASKMGR) is created during instance startup. It starts various scheduled system and user defined tasks and runs with the settings:

- Username = TASKMGR
- Routine = %SYS.TaskSuper.1
- Operating System Username = TASKMGR

2.2 ECP Server Processes

ECP server processes that are dynamically started have a routine name beginning with "ECP". The user name or Operating System Username is usually `Daemon` or `%System`, but it may be the name of the Instance Service user on Windows. Examples of process names follow:

- ECPCliR – ECP client reader
- ECPCliW – ECP client writer
- ECPSrvR – ECP server reader
- ECPSrvW – ECP server writer

2.3 Shadowing Processes

Shadowing data from a source database to a target database on a Caché shadow server is performed by a number of processes on each system. These processes begin when shadowing is configured and active. A list of processes that provide shadowing support follows, identified by the contents or the Routine column in the process display:

- SHDWSBLK – Runs on the instance which is the source of data to be shadowed on the remote shadow server. Username is `%System`, and the device is a TCP device with the SuperServer port.
- SHDWCBLK – Runs on the instance where the shadow server receives the source data and replicates it. The username is`Daemon`, and the device of the primary shadow server process is `|TCP|Port`, where port is the SuperServer port of the data source instance. The device for other shadow server processes will be the NULL device name specific to the operating system.

2.4 CSP Server Processes

CSP server processes are dynamically started. They will display CSPSystem in the `User` column when they are idle and waiting for a task. When they are active, they will display the Caché user for the CSP session and the current routine name.

The OS Username column will display CSP Gateway beginning in Caché version 2015.2 and display CSPSystem in earlier versions.

- %SYS.cspServer – the CSP server process
- %SYS.cspServer2 – the second CSP server process

For each of these servers, on Windows the executable is CSPAP.dll; on UNIX®, it is CSPap.so. The operating system username is CSP Gateway. The program name may change as the process changes tasks.

2.5 Mirroring System Processes

Mirror system processes are started if mirroring is configured. They perform various functions related to mirroring.

- MIRRORMGR – Mirror Master. The User is the description of the mirror function performed: Mirror Master, Mirror Primary, Mirror Dejournal, Mirror Prefetch, or Mirror JrnRead. The operating system username is Daemon. No TCP port is open. the device is the operating system null device.
- MIRRORCOMM – Mirror communication process. The username is Mirror Arbiter, Mirror Backup, or Mirror Svr:RdDmn. The operating system username isDaemon. The device is |TCP|XXX. The TCP port can be determined from the Device name or the Mirror configuration.

3 IP Protocols

3.1 TCP

A Caché instance accepts connections on TCP/IP ports specified by configuration options. Any Operating System restrictions on usage of ports, for example with a fire wall, require port settings to allow inbound access that are consistent with the ports configured for Cache/Ensemble. If the firewall defines rules for executables, as it does on Windows, you may need to grant permission to programs as well, for example the cache.exe, clmanager.exe, ISCAgent.exe, and the httpd.exe executables will require such permissions.

TCP/IP ports used by Caché are defined by the instance configuration. The configured ports can be examined in the cache.cpf file in the installation directory. The [Startup] section configures DefaultPort, DefaultPortBindAddress and WebServerPort. DefaultPort specifies the port on which the SuperServer accepts connections; the default value is 1972. DefaultPortBindAddress optionally specifies an interface address the SuperServer binds to. WebServerPort specifies the port on which the private web server accepts connections; the default value is 57772.

The private web server is mostly used in development environments and is not recommended for production environments.

The [SQL] section contains JDBCGatewayPort which defines the Java Database Connectivity (JDBC) gateway port number. Its default value is 62972.

The [Telnet] section contains a Port value to specify the port on which Caché telnet service (ctelnetd.exe) accepts telnet connections to Caché on Windows. The Zen reports facility may create long running HotJVM java process which will accept requests on the port configured in the Management Portal > System Administration > Configuration > Zen Reports > Settings. Render Servers, Print Servers, and Excel Servers can also be configured to listen on ports specified from Management Portal > System Administration > Configuration > Zen Reports > Settings.

3.2 UDP

Caché and the license server (clmanager or clmanager.exe) communicate primarily using the UDP protocol. Caché sends messages as UDP packets to the license server port. This port is 4001 by default, and is configured in the Management Portal > System Administration > Licensing > License Servers. The license server replies to Caché at the port Caché used to send the original message (it looks up the port in the packet header). TCP is only used between Caché and the license server during a query request. Caché opens a TCP port for accept/listen and sends this port number in the query request. The license server connects back to that port and sends the results over the TCP connection. The port number is random and is open only during transmission of the query results.

3.3 SNMP

The %System_Monitor Service enables Caché (or Ensemble) to act as a subagent to an SNMP Agent on the managed system. This supports both SNMP requests (GET or GETNEXT) for Cache/Ensemble management and monitoring data (as defined in the supplied MIBs), and SNMP Traps (asynchronous notifications sent by Cache/Ensemble). Disabling the %System_Monitor service will disable all communication between Cache/Ensemble and the SNMP Agent on the local system, and consequently with any remote SNMP manager applications.

3.4 HTTP

Refer to the description of the components of the CSP Gateway used by Caché to serve HTTP requests by navigating through the online documentation as follows: Documentation > Caché Web Development > CSP Gateway Configuration Guide > Introduction to the CSP Gateway. The private Web Server is httpd.exe (httpd on UNIX®) located in the httpd\bin subdirectory under the installation directory. Startup of the private web server is controlled by the Management Portal > System Administration > Configuration > Additional Settings > Startup > WebServer set to true or false.

3.5 Gateways

Cache provides a number of Gateways to external data. These include SQL Gateway, JDBC Gateway, Object Gateways, and XSLT 2.0 Gateway servers. The TCP/IP ports used are defined using the gateway setup pages accessed via the Management Portal > System Administration > Configuration > Connectivity. See the documentation of these gateways for an explanation of Operating System services or processes on which they depend.

4 Removing Unneeded Caché Processes

Caché service processes are not created unless the services they support are enabled and configured. There is no need to take any additional action to prevent Caché service processes from running.

5 External (non-Caché) Processes

A Caché instance will start processes running executables other than cache[.exe] to perform a number of functions in support of the instance. Instance specific versions of these executables, which are generally specific to the instance version, live in the `bin` subdirectory of the installation directory. Executables that may be shared by multiple Caché instances live in a common directory.

Persistent processes may be running the following executables, which live in the `bin` directory on Windows.

- Cache[.exe] – The Caché executable.
- clmanager.exe – The Caché license server.
- cservice.exe – The Caché Windows service. All Caché instance processes are descendants of the instance service.
- CStudio.exe – The Caché studio.
- csystray.exe – The Caché cube in the system tray.
- ctelnetd.exe – The Caché telnet server. Accepts telnet connections and creates Caché server processes to serve the telnet connection.
- CTerm.exe – The Terminal.
- ctrmd.exe – The local Terminal connection daemon. Accepts local Terminal connections (not telnet) and creates Caché server processes to serve the connection.
- cwdimj.exe – Processes the WIJ file during Caché startup and shutdown.

Persistent processes may be running the following executables, which live in the `bin` directory on UNIX®.

- cache - The Caché executable.
- clmanager – The Caché license server
- cwdimj – Processes the WIJ file during Caché startup and shutdown.

Zen Reports launches java applications which may be long running. These applications run in the java virtual machine and run Java programs under the `lib` subdirectory of the Caché installation directory. Applications include PrintServer, RenderServer, ExcelExporter and QueuingRenderServer. The purpose of PrintServer is to support printing of PDFs on Windows Vista and above. The purpose of ExcelServer is to speed up rendering of Excel ZEN Reports. The purpose of RenderServer is to speed up the rendering of ZEN Reports PDFs. Speed up is done by keeping the JVM (Java Virtual Machine) hot, so it does not have to be restarted. Cache communicates with these servers via TCP. The ports are configured with Management Portal > System Administration > Zen Reports. The SMP starts the PrintServer. The other servers are started when first Microsoft Excel or PDF report is rendered.

Other programs in the `bin` directory are used from time to time, but the processes are short running and unlikely to be displayed by a process listing for long.

Executable binaries shared by Caché instances reside in subdirectories of C:\Program Files (x86)\Common Files\InterSystems on Windows. The processes may be seen running these executable binaries from the common directory on Windows.

- ISCAgent.exe – Controls mirror failover.
- Cterm.exe – The Terminal.

Shared binaries are usually installed in /usr/local/etc/cachesys on UNIX®.

- ISCAgent* - Controls mirror failover.

In addition to executable binaries, a number of shared library binaries are stored in the common directory.

6 Ensemble

6.1 Adapters

Ensemble provides communication with external interfaces using adapters.

6.1.1 Email

Ensemble Email adapters are Caché processes. They use TCP/IP to send/receive email from an email server. Outbound adapters send mail to a SMTP server. Inbound adapters poll for relevant (filtered) messages from a POP3 serve. Email servers are likely to be on a remote server, so while there would be no local process, the remote system would need to be reachable through a firewall

6.1.2 File

Ensemble File Input Adapters are Caché processes. They periodically inspect a directory they have been configured to monitor, read files that appear there, pass the files to the Business Service they have been configured to support, and move the files to the configured archive directory. The EnsLib.File.InboundAdapter class provides the implementation. The *FilePath*, *WorkPath*, and *ArchivePath* properties define the input, temporary work, and archive directories, respectively.

Ensemble File Output Adapters are employed by Ensemble Business Operations to write data to files. The file path and file name are specified by the Business Operation and operations on the file are invoked by calling methods of the EnsLib.File.OutboundAdapter class. Messages are usually queued to a worker job that performs the actual output operation. This implies the existence of Ens.Queue processes.

6.1.3 FTP

Cache acts as a client for FTP communication with remote FTP servers using the %Net.FtpSession class. The %Net.FtpSession class can be configured to use PASV for the data channel to avoid an inbound connection. Ensemble provides FTP inbound and outbound adapters. Both act as FTP clients to get (input) or put (output) under the control of a Business Service created by the customer. The FTP server and port are configurable. The FTP adapters are Caché processes.

6.1.4 HTTP

The HTTP adapters (EnsLib.HTTP.InboundAdapter and EnsLib.HTTP.OutboundAdapter) enable productions to send and receive HTTP requests and responses. HTTP adapters are implemented by Caché processes. The port and interface IP addresses of the inbound HTTP adapter are configurable. The server and port to which the outbound HTTP adapter is provided by class settings.

6.1.5 Java Gateway

Ensemble adapters use the Java Gateway to communicate through a Java intermediary process. A java process is started which depends on the existence of a Java Virtual Machine. The Caché server process communicates with the Java process via a TCP connection. The TCP ports used are configurable.

6.1.6 LDAP

The EnsLib.LDAP.OutboundAdapter can be used like other adapters by Business Services to send requests to an LDAP server and receive responses.

6.1.7 MQSeries

Ensemble EnsLib.MQSeries.InboundAdapter and EnsLib.MQSeries.OutboundAdapter enable Ensemble productions to retrieve messages from and send messages to message queues of Ensemble IBM WebSphere MQ. Dynamically loaded shared library binaries are used for the communication.

6.1.8 Pipe

Ensemble EnsLib.Pipe.InboundAdapter and EnsLib.Pipe.OutboundAdapter enable Ensemble productions to invoke operating system commands or shell scripts. They create a process external to Caché and communicate with it via a pipe, so an

external process will exist while the Pipe adapter is communicating with it. The command the process runs is determined by the value assigned to the *CommandLine* property of the adapter class.

6.1.9 SAP

The Java Gateway is used to communicate with the SAP Java Connector using classes imported with the EnlLib.SAP.BootStrap class ImportSAP method.

6.1.10 SQL

The Ensemble SQL inbound and outbound adapters enable Ensemble productions to communicate with JDBC or ODBC-compliant databases. In general, the inbound SQL adapter (EnsLib.SQL.InboundAdapter) periodically executes a query and then iterates through the rows of the result set, passing one row at a time to the associated business service. The SQL adapters use the underlying capabilities of Caché SQL and JDBC Gateways.

6.1.11 TCP

Ensemble provides input and output TCP adapters. Each TCP inbound adapter checks for data on a specified port, reads the input, and sends the input as a stream to the associated business service. Within a production, an outbound TCP adapter is associated with a business operation that you create and configure. The business operation receives a message from within the production, looks up the message type, and executes the appropriate method in the outbound TCP adapter to transmit the data over TCP.

6.1.12 Telnet

Ensemble provides the EnsLib.Telnet.OutboundAdapter which permits outbound telnet connections to the telnet facility on another system. This adapter provides methods to programmatically emulate the effect of manually logging into the remote system using telnet client software. The Caché TCP device is the underlying technology.

7 Checklist for Hardening Your Deployment

This checklist is intended to provide your organization with guidelines for assessing how secure your environment is and to provide tips for hardening your environment that will help your organization avoid and prevent security breaches. This checklist is not intended to be a “how to list” and is not all-inclusive. The points below are items to consider rather than a definitive list of rules to apply.

You alone are responsible for the security of your infrastructure. If you are uncertain about your approach to hardening and protection, consult a security professional.

7.1 Network and Firewalls

ID	Topic	Description
1.	Network, hardware, software and policies	Obtain copies of and review security polices, firewall logs, firewall configuration and patch levels, public facing IP addresses, diagrams of network, and firewall topologies.
2.	Audit physical environment	Ensure firewalls and management servers are in a physically secure location that can only be accessed by authorized personnel; and that they are patched up to date.
3.	Review change management process, rule base modifications	Review procedures and approval process for changes. Automation tools are available for this.
4.	Run vulnerability tests	Run automated tools to analyze and identify insecure services, protocols, and ports.
5.	Use brute force detection systems	Stop people from guessing passwords, and prevent them from connecting to the server, by blocking their current IP address in your server firewall.
6.	Ongoing Audits and real-time monitoring and alerting	Ensure a process is in place for continuous auditing of firewalls. Ensure real-time monitoring is in place to alert on changes to the firewall. Review their logs regularly.

7.2 Operating System

ID	Topic	Description
1.	Plan the installation	Understand the server role, and document the install procedure. Download appropriate operating system securing and hardening guides for more detailed information.
2.	Patch levels	Ensure operating system patches are up to date, especially security patches. Turn off automatic updates.
3.	Anti-virus	Install this software where appropriate, that is Windows servers and client PCs.
4.	Disable unnecessary software, services & ports	Disable unnecessary network services such as IPv6, telnet, and FTP. Disable unnecessary daemons that are not used such as DHCP, scheduling and queuing services, and Laptop services. Configure in-use services to be as secure as possible; for example, secure SSH by limiting SSH protocol to Version 2 (Version 1 is not secure).
5.	Logs	Maintain server logs and mirror those logs to a separate log server.
6.	Monitoring and Alerting	Configure monitoring and alerting settings to notify of events such as changes to the system, and unauthorized access.
7.	Physical Security	Configure the BIOS to disable booting from CDs/DVDs, floppies, and external devices; set a password to protect these settings.

7.3 Web Server

ID	Topic	Description
1.	Plan the installation	Understand the role of the web server: what content will it serve; will the pages be static; what web services are provided? Document the installation procedure. Download and review the appropriate hardening security guide.
2.	Patch levels	Ensure web server is up to date, especially with regard to security patches.
3.	Web server header info	Configure the servers so that HTTP headers do not provide information relating to the web server software being run, or system types and versions.
4.	Turn off HTTP TRACE	When enabled, HTTP TRACE request is used to echo back all received information.
5.	Error handling	Implement proper error handling by utilizing generic error pages and error handling logic to force the application to avoid default error pages. These often leak sensitive system and application information.
6.	Disable modules	Disable all unused modules to reduce surface area of the web server; these modules often provide too much information – *Apache:* autoindex, cgi, imap, info, status, userdir, actions, negotiation… *IIS:* ASP, ASP.NET, WebDAV, CGI, directory browsing…
7.	Users and Groups	*Apache:* Run Apache as separate user and group so Apache processes cannot be used by other system processes. *IIS:* Remove unused accounts, disable Guest account

7.4 Users, Passwords, Groups, Ownerships, and Permissions

ID	Topic	Description
1.	User management	Disable root login. All administrators should be named users. Regularly check for unused user accounts, and for default user accounts and passwords.
2.	Password policy	Require and use very strong passwords with mixed case, numbers, and special characters. Change passwords on a regular basis. Lock accounts after too many login failures.
3.	Unix	Create groups and users before installation (cachemgr and cacheusr). Caché must be installed as root. Ensure groups, ownerships and permissions for Caché databases are maintained as specified.
4.	Windows	Caché must be installed using the Windows Administrator. The default Windows Administrator account should then be disabled. Also disable Guest and Help Assistant accounts.

7.5 Encryption (Data At Rest and Data In Motion)

ID	Topic	Description
1.	Data At Rest	Ensure all production data at rest on disk is encrypted.
2.	Key Management	Review the key management policies and procedures.
3.	Data In Motion	Ensure all HTTP data communications is encrypted, such as with SSL/TLS. Check SSL/TLS configuration is using latest versions of SSL/TLS.

7.6 Caché Security

ID	Topic	Description
1.	Installation	Always install with Caché Security type Locked Down.
2.	Authentication	Regularly review users and passwords.
3.	Authorization	Review application requirements; and define roles, resources and services.
4.	Auditing	Ensure Caché auditing is enabled. Review the logs regularly.
5.	Disable Services	If services such as ECP, shadowing, mirroring are not used, do not enable them.
6.	Remove unused databases and applications.	Remove unused databases such as SAMPLES and USER.