# Table of Contents

**About This Book** ................................................................................................................................. 1  
General Licensing Notes .................................................................................................................. 1  
Application Use Of InterSystems Web Server ................................................................................. 1  

**1 New and Enhanced Features for Caché 2016.2** ............................................................................. 3  
1.1 Major Developments .................................................................................................................. 3  
1.1.1 Support for Atelier, New Eclipse-Based IDE ................................................................... 3  
1.1.2 Support for OAuth 2.0 and OpenID Connect ................................................................... 3  
1.1.3 SQL/JSON Support .......................................................................................................... 4  
1.1.4 SQL Improvements .......................................................................................................... 4  
1.2 Other Items of Note .................................................................................................................... 4
About This Book

This book provides information on the major features that have been added to Caché in this release.

It contains the following section:

- Caché 2016.2 Release Notes

The following books provide related information:

- Caché Upgrade Checklists discusses changes made to each release that may adversely affect existing Caché applications.
- Caché Release Notes and Upgrade Checklist Archive provides release notes describing new features for previous releases and the issue checklist for version prior to those covered in the Caché Upgrade Checklists.
- InterSystems Supported Platforms lists the technologies supported by this release of Caché.
- Caché Installation Guide describes the process of installing Caché on your system.
- Introduction to Caché provides an overview of the features and major components of Caché.

General Licensing Notes

InterSystems makes its products and features available under license to customers. While InterSystems may or may not enforce the use of said products or features consistent with the purchased license capabilities, customers are expected to comply with terms of their licenses. Moreover, InterSystems may tighten enforcement at any release without notice.

Developers must be aware that certain license types are required in order to use specific product features such as Multi-Server capability, Mirroring, and Web Services features. The specific requirements are noted in the InterSystems Price List and the Terms and Conditions for licensing. These are available from your local InterSystems representative.

Application Use Of InterSystems Web Server

InterSystems installs an Apache-based web server (often referred to as the "private web server") to assure that the management portals for its products are always available. The private web server is built and configured to meet the management needs of InterSystems administrative servers and is configured to only connect to InterSystems products. The options selected are not, in general, suitable for production use - in particular, security is minimal and the options used are generally unsuitable for a high volume of HTTP requests. Testing, by InterSystems, of the private webserver only covers use of the private web server for managing Caché, Ensemble, HealthShare, and other InterSystems products.

Customers are not required to use this web server to manage our products. You may also use a web server of your choice, located on whatever server you elect to use. The private web server is provided as a convenience to simplify installation and installation dependencies. Many developers also find it useful to use the private web server for unit testing.

UNIX®

The parameters used for the UNIX® build are:

- --prefix=<installation_location>
- --disable-actions
- --disable-asis
The server produced has defaults using the Apache Group’s prefork Multi-Processing Module (MPM). This is the non-threaded server model. The number of requests that can be concurrently served is directly related to the number of Apache worker processes in the pool. The private web server is configured to occupy the smallest possible footprint by allowing a maximum of two worker processes to be created for the pool. The following settings will be found in the Apache configuration (httpd.conf) for the server:

- MinSpareServers: 1
- MaxSpareServers: 2

By contrast, the default Apache configuration for a production grade build is usually the following:

- StartServers: 5
- MinSpareServers: 2
- MaxSpareServers: 20
- ServerLimit: 256
- MaxClients: 256

This configuration will allow Apache to create 5 worker processes at start-up time, increasing to a maximum of 256 as the concurrent load increases. Because of these differences in configuration, the performance of the private web server will be noticeably inferior to that of a production grade Apache build as the concurrent load increases.

Windows

Windows-based Apache installations use the official binary distribution for Windows and a special multithreaded Multi-Processing Module (MPM) which is more suited to the way the operating system is optimized. However, since InterSystems installs and loads only a small subset of modules (mod_alias.so, mod_authz_host.so, mod_log_config.so and mod_mime.so.), their functionality is limited.

Conclusion

If you expect very low volume of HTTP traffic, have limited demands for high availability and secure operation, the private web server may be suitable for your deployment needs. However, if you expect a high amount of HTTP traffic, require high availability in your webserver, need to integrate with other sources of web information, or need a high degree of control over your web server, InterSystems recommends installing your own separate copy of Apache, ideally on its own server, and configuring it to use our CSP gateway to communicate with Cache, Ensemble, or HealthShare. Review the options above to determine if this is so.

Note: This applies to Unix®, Linux, Mac OS X and Windows systems. A private web server is not installed on OpenVMS.
1

New and Enhanced Features for Caché 2016.2

This section includes:

• Major Developments
• Other Items of Note

1.1 Major Developments

The following major, new features have been added to Caché for this release:

• Support for Atelier, New Eclipse-Based IDE
• Support for OAuth 2.0 and OpenID Connect
• SQL/JSON Support
• SQL Improvements

1.1.1 Support for Atelier, New Eclipse-Based IDE

This is the first Caché release that provides server support for Atelier, the new Eclipse-based IDE. The Atelier IDE brings together the powerful and popular Eclipse development environment and the InterSystems Caché database. Atelier allows you to develop Caché applications using a modern file-based IDE on a client system. Atelier handles uploading the application to the Caché server where it can be run and debugged.

The focus of future development will be on the new Eclipse based IDE. Studio will remain an option to install and developers can continue to develop code with it. However, it will be treated as a maintenance product and will not see new functionality added as we move forward with Atelier. Some minor bugs may not be addressed either depending on resources required versus the severity of the issue.

1.1.2 Support for OAuth 2.0 and OpenID Connect

Starting with this release, Caché supports the OAuth 2.0 framework and OpenID Connect Core. Caché can act as a client, as a resource server, and as an authorization server. See Using OAuth 2.0 and OpenID Connect with Caché. Also see this
New and Enhanced Features for Caché 2016.2

The SQL syntax has been enriched to support parts of the SQL/JSON standard. This allows to construct JSON data as a result from an SQL query and also to query JSON data. The latter requires a mapping from JSON paths to columns and types to allow usage in a relational environment. The functions supported are:

- **JSON_OBJECT**—construct a JSON object for each encountered row.
- **JSON_ARRAY**—construct a JSON array for each encountered row.

**1.1.4 SQL Improvements**

This release has the following SQL performance enhancements and improvements:

- **%Parallel**—Improves performance of queries by breaking the query into chunks that can be executed in parallel on multiple processor systems. In this release the %Parallel infrastructure has the following improvements:
  - Lower overhead to breaking a query into chunks.
  - Caché dynamically breaks the queue into a number of chunks based on the number of hardware threads that are available.
  - Increased number of query conditions that can be broken into chunks. For example, Caché can now break set looping constructs like IN (?, ?, ?) or %INLIST() into chunks.

The result of these changes is that you can add %Parallel to most queries without extensive analysis. If Caché can improve the performance by breaking the task into chunks and using parallel processing, it will. If the query is not amenable to parallel processing or if the system only has a single hardware thread available, Caché recognizes this situation and does not incur the overhead of breaking the query into chunks.

- Improved performance when iterating over list collections.
- **Frozen plans**—allows you to freeze the plans while you make index and other changes.

**1.2 Other Items of Note**

In addition, many more lesser improvements and corrections are also included. In particular, if you are upgrading an existing installation, please review the detailed list of changes in the Upgrade Checklist.

Other minor areas of improvement include:

- **New REST API to access Caché source code files is available in this release.** This API was implemented to provide access to the Atelier IDE, and it allows you to create IDEs and similar application. See *Accessing Caché Source Code Files Using REST* for details.

- The DataCheck utility makes more efficient use of parallel processing in order to complete checks faster. The ^DATACHECK operator interface has an enhanced status page and start dialog for improved usability.

- Caché is enhanced to use faster memory synchronization mechanisms (memory barriers) in certain critical execution paths. Data ingestion tests show more than 10% improved performance on Linux x86-64. The improvements apply to the following platforms.
- Apple Mac OS X for x86–64
- IBM AIX® for Power System-64
- Microsoft Windows for x86–32, x86–64
- Linux for x86–32, x86–64
- Solaris for x86–64, SPARC-64

These improvements do not affect the Itanium and Alpha platforms.