



# Caché Glossary of Terms

Version 2018.1  
2024-11-07

*Caché Glossary of Terms*

PDF generated on 2024-11-07

InterSystems Caché® Version 2018.1

Copyright © 2024 InterSystems Corporation

All rights reserved.

InterSystems®, HealthShare Care Community®, HealthShare Unified Care Record®, IntegratedML®, InterSystems Caché®, InterSystems Ensemble®, InterSystems HealthShare®, InterSystems IRIS®, and TrakCare are registered trademarks of InterSystems Corporation. HealthShare® CMS Solution Pack™, HealthShare® Health Connect Cloud™, InterSystems® Data Fabric Studio™, InterSystems IRIS for Health™, InterSystems Supply Chain Orchestrator™, and InterSystems TotalView™ For Asset Management are trademarks of InterSystems Corporation. TrakCare is a registered trademark in Australia and the European Union.

All other brand or product names used herein are trademarks or registered trademarks of their respective companies or organizations.

This document contains trade secret and confidential information which is the property of InterSystems Corporation, One Memorial Drive, Cambridge, MA 02142, or its affiliates, and is furnished for the sole purpose of the operation and maintenance of the products of InterSystems Corporation. No part of this publication is to be used for any other purpose, and this publication is not to be reproduced, copied, disclosed, transmitted, stored in a retrieval system or translated into any human or computer language, in any form, by any means, in whole or in part, without the express prior written consent of InterSystems Corporation.

The copying, use and disposition of this document and the software programs described herein is prohibited except to the limited extent set forth in the standard software license agreement(s) of InterSystems Corporation covering such programs and related documentation. InterSystems Corporation makes no representations and warranties concerning such software programs other than those set forth in such standard software license agreement(s). In addition, the liability of InterSystems Corporation for any losses or damages relating to or arising out of the use of such software programs is limited in the manner set forth in such standard software license agreement(s).

THE FOREGOING IS A GENERAL SUMMARY OF THE RESTRICTIONS AND LIMITATIONS IMPOSED BY INTERSYSTEMS CORPORATION ON THE USE OF, AND LIABILITY ARISING FROM, ITS COMPUTER SOFTWARE. FOR COMPLETE INFORMATION REFERENCE SHOULD BE MADE TO THE STANDARD SOFTWARE LICENSE AGREEMENT(S) OF INTERSYSTEMS CORPORATION, COPIES OF WHICH WILL BE MADE AVAILABLE UPON REQUEST.

InterSystems Corporation disclaims responsibility for errors which may appear in this document, and it reserves the right, in its sole discretion and without notice, to make substitutions and modifications in the products and practices described in this document.

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](mailto:support@InterSystems.com)

# Table of Contents

<b>About This Book .....</b>	<b>1</b>
<b>1 Complete Glossary of Terms .....</b>	<b>3</b>
1.1 Terms Beginning with Symbols and A .....	3
1.2 Terms Beginning with B .....	5
1.3 Terms Beginning with C .....	6
1.4 Terms Beginning with D .....	13
1.5 Terms Beginning with E .....	16
1.6 Terms Beginning with F .....	18
1.7 Terms Beginning with G .....	19
1.8 Terms Beginning with H .....	19
1.9 Terms Beginning with I .....	20
1.10 Terms Beginning with J .....	22
1.11 Terms Beginning with K .....	22
1.12 Terms Beginning with L .....	23
1.13 Terms Beginning with M .....	24
1.14 Terms Beginning with N .....	26
1.15 Terms Beginning with O .....	27
1.16 Terms Beginning with P .....	29
1.17 Terms Beginning with Q .....	32
1.18 Terms Beginning with R .....	32
1.19 Terms Beginning with S .....	35
1.20 Terms Beginning with T .....	39
1.21 Terms Beginning with U .....	39
1.22 Terms Beginning with V .....	40
1.23 Terms Beginning with W .....	41
1.24 Terms Beginning with X .....	41
1.25 Terms Beginning with Z .....	42



# About This Book

This book defines terms relevant to Caché. It lists them alphabetically, beginning with symbols such as \$. Its topics are:

- [Terms Beginning with Symbols and A](#)
- [Terms Beginning with B](#)
- [Terms Beginning with C](#)
- [Terms Beginning with D](#)
- [Terms Beginning with E](#)
- [Terms Beginning with F](#)
- [Terms Beginning with G](#)
- [Terms Beginning with H](#)
- [Terms Beginning with I](#)
- [Terms Beginning with J](#)
- [Terms Beginning with K](#)
- [Terms Beginning with L](#)
- [Terms Beginning with M](#)
- [Terms Beginning with N](#)
- [Terms Beginning with O](#)
- [Terms Beginning with P](#)
- [Terms Beginning with Q](#)
- [Terms Beginning with R](#)
- [Terms Beginning with S](#)
- [Terms Beginning with T](#)
- [Terms Beginning with U](#)
- [Terms Beginning with V](#)
- [Terms Beginning with W](#)
- [Terms Beginning with X](#)
- [Terms Beginning with Z](#)

For information about fundamental Caché topics, see the following documents:

- [\*Caché Programming Orientation Guide\*](#)
- [\*Introduction to Caché\*](#)
- [\*Using Caché Objects\*](#)
- [\*Using Caché SQL\*](#)
- [\*Caché System Administration Guide\*](#)

- [\*Caché Security Administration Guide\*](#)

For general information, see [\*Using InterSystems Documentation\*](#).

# 1

## Complete Glossary of Terms

The following is a list of terminology relevant to Caché.

### 1.1 Terms Beginning with Symbols and A

#### **\$X/\$Y action table**

*Caché System.* Internal table that specifies whether a character, on input to or output from a device, modifies \$X, \$Y, or both. The special variables \$X and \$Y keep track of the horizontal and vertical coordinates on the device.

#### **\$ZF function**

*ObjectScript.* An InterSystems implementation-specific function that lets you invoke external programs or routines from within Caché. On UNIX®, you can use \$ZF functions to issue UNIX® shell commands, call UNIX® System Services, or call routines written in high-level languages, such as C.

#### **\$ZU function**

*ObjectScript.* A group of system-supplied functions, some of which were formerly made externally accessible. Deprecated; removed from Caché documentation and replaced by classes as of the 2010.2 release.

#### **ASCII character set**

*Caché System.* For InterSystems products, ASCII refers to extended, 8-bit character sets, rather than the more limited, 7-bit character set. By default, a Caché instance uses the character set from the machine's locale, as specified in the “Supported Languages” chapter of the online [InterSystems Supported Platforms](#) document for this release; if there is no support for a machine's default language, Caché uses the ISO-8859-1 character set, commonly known as Latin-1.

#### **abstract class**

*Caché Objects.* An abstract class cannot be instantiated. It acts as a template for multiple non-abstract subclasses with common characteristics.

#### **abstract persistent class**

*Caché Objects.* An abstract persistent class cannot be instantiated but is projected to Caché SQL as a table containing all of the data stored in its subclasses.

## **access mode**

*Caché System.* A category of the available connection tools, where each access mode has particular characteristics and is for particular situations. There are three different access modes: local (where the user interacts directly with Caché on the server machine); client/server (where the user connects to Caché from a separate executable, usually on a separate machine); and CSP (where the user interacts with Caché through a CSP web-based application).

## **activate**

*Caché System.* To retrieve a database-encryption key from memory and store its value internally in a running Caché instance so that Caché can mount and manipulate an encrypted database.

## **ActiveX**

*General.* A set of technologies that enables software components to interact with one another in a networked environment, regardless of the language in which the components were written.

## **added role**

*Caché System.* Any role that becomes associated with a user after that user authenticates to Caché.

## **ad-hoc query**

*General.* A SELECT statement used in embedded SQL or via ODBC or JDBC. Within object applications, ad-hoc queries differ from normal queries in that they are not part of a class definition and they cannot be processed using the query interface.

## **application mode**

*Caché System.* The mode in which Caché initiates a terminal session in a pre-specified namespace and running a pre-specified program. In application mode, the user cannot initiate programs from the Caché prompt, and the terminal session ends when the user shuts down the program. Application mode is designed to allow users to perform a limited set of operations and not to gain access to other, more powerful aspects of the Terminal that are available in [programmer mode](#).

## **Application Program Interface (API)**

*General.* A set of calls that provide an interface between an application tool or C application and the server and that manage communications with the server.

## **array**

*Caché Objects.* A collection that uses key-value pairs to access data. Arrays are projected to SQL as child tables. In Caché Basic, a set of sequentially indexed elements where each element of an array has a unique identifying index number. Changes made to one element of an array do not affect the other elements. Caché Basic sometimes uses the term “persistent multidimensional array” to refer to [globals](#).

## **asset**

*Caché System.* Data or functionality that is protected by Caché security. Assets are represented within Caché as resources; each resource represents one or more assets.

## **atomic lock**

*Caché Objects.* An atomic lock performs no locking for data stored in a single node of the database. It holds a shared lock while data stored in more than one node is being loaded and holds an exclusive lock while the same data is saved.



**attribute**

*Caché Objects.* A data element representing a specific characteristic associated with a class. Each object of that class contains a literal value for the attribute. Together, the attributes of a class represent its state. Also known as a property. Strictly speaking, an attribute is any property that is not a relationship.

**audit database**

*Caché System.* A database containing a list of some of the events that have occurred while Caché is running. When auditing is enabled, Caché logs various events, depending on the choices of the Caché administrator. Applications can also create and log their own events. The audit database is also known as the audit log.

**authentication**

*Caché System.* The process by which users demonstrate that they are who they say they are. In the most commonly recognized case, this occurs by providing a password. However, other authentication schemes are commercially available.

**authentication mechanism**

*Caché System.* A means by which a user is authenticated to Caché. Authentication can occur using Kerberos, the use of existing operating system credentials, a native Caché password prompt, LDAP, or delegated (that is, user-defined) code.

**authorization**

*Caché System.* The process of determining what an authenticated user can do on a Caché system. Authorization includes several aspects: assigning roles to users (performed by a [role-assignment mechanism](#)); managing those roles and what they can do; and managing resources within Caché.

## 1.2 Terms Beginning with B

**BLOB**

*Caché SQL.* A BLOB, or Binary Large Object, is the relational term for a large chunk of data. Caché uses its stream interface to manipulate BLOBs.

**Basic**

*Caché System.* One of the programming languages supported by Caché.

**binary stream**

*Caché Objects.* A binary stream provides an interface to manipulate and store large chunks of binary data such as images. Data stored in a binary stream is not translated during Unicode conversions. The Caché stream interface can be used to manipulate binary streams in ObjectScript, SQL, ActiveX, and Java.

**bind**

*Caché SQL.* To associate a variable in an application program with a host variable using embedded SQL.

**breakpoint**

*ObjectScript.* Location in a Caché routine that you specify with the [ZBREAK](#) command. When execution reaches that line, Caché suspends execution of the routine and, optionally, executes debugging actions you define.

## **buffer**

*General.* Defined location of computer memory that holds specific information. Buffers make it easy for multiple processes to access the same data. For instance, Caché caches globals in global buffers, and routines in routine buffers.

# **1.3 Terms Beginning with C**

## **CACHE.DAT**

*Caché System.* The primary volume in a Caché database. It contains Caché globals and routines.

## **CACHE.EXT**

*Caché System.* A secondary volume in a Caché database. A Caché database can contain from 0 through 7 CACHE.EXT files, in addition to one CACHE.DAT file. A CACHE.EXT file contains Caché globals and routines, but cannot be referenced directly except when the Databases or MSU utility is used to add it to the database.

## **CACHE.KEY**

*Caché System.* The file that contains the encoded version of your Caché license information.

## **CLASSPATH**

*Java.* A system variable used by Java compilers to search for classes and packages referenced in the classes they compile.

## **CZF.M64**

*Caché System.* A special assembly language file included on your Caché distribution tape or DVD, that helps you invoke routines written in languages other than ObjectScript from your Caché programs. A C language version of this file, CZF.C, allows you to invoke C routines only.

## **Caché cube**

*Caché System.* The icon that appears on the taskbar when you start Caché on a Windows system. From this icon, you can [configure and manage your Caché systems](#) as well as create and manage classes and routines.

## **Caché database**

*Caché System.* A collection of related data stored in globals and routines within a single directory, namespace, or UIC.

## **Caché list control**

*Caché Objects.* An ActiveX control written for Visual Basic that aids in the display of query results. The Caché List Control does not provide the interface for selecting queries and query parameters for execution, just a way to display the results.

## **Caché login**

*Caché System.* The native Caché authentication system: a user is prompted for a password, a hash of the supplied password is passed to the Caché server, and that hash is compared to the hash of the existing password stored in the server. If the two values are identical, Caché then grants authorization to those resources for which the user has privileges.

This mechanism is listed in the Management Portal as **Password** authentication. Since Kerberos and OS-based authentication both typically use passwords, the documentation uses the term “Caché login.”

### **Caché Object Form Wizard**

*Caché Objects.* A Visual Basic add-in that allows you to quickly and easily create a simple form to access properties of a single Caché class.

### **Caché Object Server for ActiveX**

*Caché Objects.* The Caché Object Server for ActiveX is a complete ActiveX in-process automation server that gives client applications access to server-based Caché objects.

### **Caché Object Server for Java**

*Caché Objects.* The Caché Java Binding gives client applications written in Java access to server-based Caché objects.

### **Caché object utility library**

*Caché Objects.* The Caché Object Utility Library provides an interface to configure the object components of Caché, manipulate and compile classes, and interactively use objects. The primary interface to these utilities is via the %SYSTEM.OBJ class.

### **ObjectScript**

*Caché System.* One of the programming languages supported by Caché.

### **Caché query control**

*Caché Objects.* An ActiveX control written for Visual Basic that provides a simple interface for executing queries and displaying the results. The Caché Query Control provides an interface for selecting at runtime any query that returns the ID and for specifying any query parameters.

### **Caché relational utility library**

*Caché SQL.* The Caché relational utility library provides an interface to configure Caché SQL, manage the Caché SQL Server, and import DDL from other relational databases. The primary interface to these utilities is via the %SYSTEM.SQL class.

### **Caché SQL server**

*Caché SQL.* The Caché SQL server handles SQL requests from client applications to apply queries and updates to a Caché database, error logging, and other SQL-related tasks. This is the server process that the ODBC and JDBC drivers connect to.

### **Caché SQL storage**

*Caché Objects.* Caché SQL storage provides a mechanism for using SQL to insert, update, and retrieve data via the object persistent interface. Classes use Caché SQL Storage if their storage definition specifies the %CacheSQLStorage storage class and includes an SQL mapping section.

### **Caché SQL**

*Caché SQL.* An advanced relational interface to Caché. Caché SQL is fully integrated with the object functionality of Caché, sharing its Class Dictionary for table definition and using Advanced Data Types defined as classes.

## Caché Server Pages (CSP)

*Caché System.* A set of technologies that provide the ability to build web-based applications with Caché.

## Caché server

*Caché System.* Facility that allows you to use a distributed Caché database in a networked system.

## Caché storage

*Caché Objects.* The default storage type for persistent objects. If you use Caché Storage, you do not need to specify any details about how to store your data and your classes are automatically projected to SQL.

## Terminal

*Caché System.* A terminal window that provides an interactive command line interface to Caché.

## Caché web server

*Caché System.* A private, Caché-specific web server that supports the Management Portal and the Caché online documentation. The private web server does not interfere with any other web server on the same host. The default web server port is 57772; if 57772 is not available, the web server listens the next available number higher than 57772. To set the web server port number, use the **WebServerPort** field of the **Startup Settings** page of the Management Portal (**System > Configuration > Startup Settings**) available through the **System Administration >> Configuration >> Additional Settings >> Startup Settings** choices.

## calculated property

*Caché Objects.* A property that has no in-memory storage allocated for it when the object containing it is instantiated. Instead, its value is determined each time it is requested.

## call method

*Caché Objects.* A method that directly invokes a Caché routine.

## callin interface

*Caché System.* Caché facility that lets you execute and evaluate ObjectScript commands and expressions from within C programs. You can also use the callin interface from \$ZF routines.

## callback method

*Caché Objects.* Callback methods are called by system methods to allow additional user written processing during specific events. To distinguish these types of methods, they are given names of the form **%OnEvent**, where “Event” describes the event that triggers the callback.

## canonical form

*General.* The canonical form of a name is its complete, standard, and unambiguous form. Caché has canonical forms for various entities, including numbers, global references, command options, files and paths, and devices.

For information on the canonical form of numbers, see the [\\$NUMBER](#) reference page in the *Caché ObjectScript Reference*.

For information on the canonical form of globals and the related naked global references, see either the section “[Naked Global References](#)” in the chapter “Using Multidimensional Storage (Globals)” in the book *Using Caché Globals* or the [\\$NAME](#) or [\\$ZREFERENCE](#) reference pages in the *Caché ObjectScript Reference*.

Information on the canonical form of command options appears on the reference page for the command itself.

The canonical form of files, paths, and devices depends on the operating system.

For example, on Windows, a reference to the file `..\cache.wij` is not in canonical form, while `C:\InterSystems\Cache\CACHE.WIJ` is (if there is an instance of Caché called “Cache” and installed in the standard location).

Device names have canonical forms on UNIX®, for example: `/dev/tty`.

### **cascading dot syntax**

*Caché Objects.* Cascading dot syntax is used to follow the chain of other objects referenced by an object and directly manipulate the properties and methods of these referenced objects. For example:

#### **ObjectScript**

```
Write auto.Engine.Cylinder.SparkPlug.IdNumber
```

### **character set**

*Caché System.* A set of values that defines the internal representation of Caché data for an instance.

### **character stream**

*Caché Objects.* A character stream provides an interface to manipulate and store large chunks of text-based data. Data stored in a character stream is translated during Unicode conversions. The Caché stream interface can be used to manipulate character streams in ObjectScript, SQL, ActiveX, and Java.

### **child table**

*Caché SQL.* A table that is dependent on another table (its parent). Rows in a child table must have a pointer to a row in the parent table.

### **circular reference**

*Caché Objects.* A set of two properties in different classes where each property is a reference to the other class. Caché fully supports circular references as long as both properties are not required properties. If both properties in a circular reference are required, you will not be able to save objects of either class.

### **class compiler**

*Caché Objects.* The class compiler compiles Caché classes. You can use the class compiler from the Studio or by using one of the compile calls in the Caché object utility library.

### **Class Definition Language**

*Caché Objects.* The Caché Class Definition Language, CDL, is a keyword-based language used to define classes in Caché.

### **class descriptor**

*Caché Objects.* A special type of routine that contains the runtime information (list of methods and properties) needed to use objects.

### **class dictionary**

*Caché Objects.* The Class Dictionary holds the current class definition of all compiled classes. The Class Dictionary is also used by Caché SQL to determine valid table definitions.

### **class hierarchy**

*Caché Objects.* The class hierarchy shows the chain of classes from each root class through each of its subclasses and their subclasses.

### **class member**

*Caché Objects.* An element that belongs to a class. Class members can be properties, methods, parameters, queries, indices, triggers, or XData blocks.

### **class method**

*Caché Objects.* A method that can be invoked whether or not an instance of its class is in memory.

### **class**

*Caché Objects.* A class encapsulates the state and behavior of a single entity. A class consists of some type information and a set of class members including properties, methods, parameters, queries, and indices.

Caché supports two main kinds of classes: data type classes and object classes. It supports many kinds of object classes including abstract classes, abstract persistent classes, persistent classes, embeddable classes, non-registered classes, registered classes, and system classes.

### **client data type**

*Caché Objects.* The client data type is used to specify information needed to project data to clients via a Caché Object Server. Every data type class must have a client data type. Attributes based on a data type class project that class's client data type as the attribute's data type to ActiveX and Java clients.

### **client lock**

*Caché System.* See outgoing lock.

### **client**

*Caché System.* The machine on which your application runs. It is connected to the server machine, on which your database resides. Also, a node on a network that can request information from other nodes.

### **client application**

*Caché System.* An executable separate from the Caché executable, but which relies on Caché, for example, as a data source. Such an application can be registered with Caché and thereby regulated as part of the Caché security model. The user of a client application must be authenticated to Caché using one of the supported authentication mechanisms.

### **code method**

*Caché Objects.* A method that executes ObjectScript or Basic code.

### **collation rule**

*Caché System.* An algorithm that determines the order in which Caché retrieves global nodes. Caché built-in collation rules include UNICODE, ANSI and string. In ANSI collation, canonical numeric subscripts come first in numeric order, followed by all others in string order; string order is the order of characters in the character set. String rule orders all subscripts as strings in the character set, just as the ObjectScript "]" operator does.

**collation sequence**

*Caché System.* Specifies the order in which Caché retrieves global nodes, when it must retrieve nodes in order. A character set and a collation rule together make up a collation sequence. Each global is associated with a particular collation sequence.

**collation**

*Caché SQL.* Collation specifies how Caché SQL transforms data before it is sorted. Each data type has a default collation type. Properties based on a data type use its collation type unless a property collation type is explicitly defined. Indices use the collation type of the properties they index unless an index collation type is explicitly specified.

Caché SQL supports the following preferred collation types: EXACT, SQLUPPER, and SQLSTRING. SQLUPPER is the default for data types based on %String. EXACT is the default for all other data types. The following legacy collations are also supported: ALPHAUP, MINUS, PLUS, SPACE, STRING, and UPPER.

**collection**

*Caché Objects.* A collection is a property containing multiple elements (either literal values or objects). Caché supports two types of collections: lists and arrays.

**column**

*Caché SQL.* A column contains a specific element of data for each instance stored within a table. Properties are projected to SQL as columns.

**communications protocol**

*General.* A set of conventions that define how data is transferred between computers on a network. More specifically, it is the software that determines how a message packet is formatted. Communication protocols are used to talk to a network interface device. More than one communication protocol can share the same physical interface device.

**compilation flag**

*Caché Objects.* A compilation flag tells the class compiler how to compile classes. Compilation flags control which classes are compiled, whether source code is deleted from the system, and how Caché handles compiling classes with objects in memory.

**compilation**

*Caché Objects.* The process which transforms Caché classes and routines into executable code.

**compiler directive**

*Caché Objects.* A compiler directive provides information on the files needed to successfully compile a specific class or routine. Examples of compiler directives include specifying include files and specifying that a class must be compiled after another class has already been compiled.

**computed field**

*Caché Objects.* A field whose value is derived from a calculation defined in ObjectScript compiled code. ObjectScript compiled code can reference other fields in the associated base table as well as ObjectScript functions and special variables.

**computer name**

*Caché System.* The name assigned to one processor, its peripheral equipment, and all of its associated datasets. This name must be unique across the network. See also System Name.

### **concurrency mode**

*Caché Objects.* The concurrency mode determines what type of locking is performed when you open and save objects. Caché supports five concurrency modes (0-4) which allow you to specify no locking, atomic locking, shared locking, shared retained locking, and exclusive locking.

### **concurrency**

*Caché Objects.* Concurrency provides a mechanism for controlling data integrity through locking. Caché supports five concurrency modes ranging from performing no locking to holding an exclusive lock from the time an object is loaded into memory until it is saved. By default, Caché uses atomic locking for all objects.

### **concurrent backup**

*Caché System.* A backup performed without stopping other database activity. Caché uses a multipass method that minimizes the impact of backup on users while maintaining the integrity of the backup.

Note: Transaction processing that occurs during backup may not be completely journaled if you clear, delete, or replace the current journal file as part of backup.

### **configuration**

*Caché System.* A Caché configuration describes Caché resources at startup. You define a configuration in the Management Portal. A configuration consists of a system configuration that you define in the System Configuration editor and a namespace/network configuration that you define in the National Language Support editor. You can create more than one configuration, although only one can be current at a time. Caché uses the current configuration at startup.

### **connection**

*Caché System.* A link between a client application or tool and Caché. Each connection has an associated process on the Caché server.

### **connection security level**

*Caché System.* Specifies which Kerberos functionality protects a client/server connection. The available levels are: initial authentication only (often simply called “Kerberos”); initial authentication and packet integrity (often called “Kerberos with packet integrity”); and initial authentication, packet integrity, and encryption of all messages (often called “Kerberos with encryption”).

### **connection tool**

*Caché System.* The means by which users establish their connection with Caché (such as the Terminal, Java, or CSP).

### **conversion code**

*Caché Objects.* Compiled code used by Caché to convert data values from internal storage formats to external display formats, and from external input formats to internal storage formats.

### **credentials cache**

*Caché System.* A file containing authentication information for a previously identified user. The credentials cache is used to speed up the authentication process by avoiding duplicated effort. Used most frequently in reference to Kerberos authentication.



**CSP application**

*Caché System.* A set of Caché Server Pages classes or pages that, together, create a Web-based application. Such an application can be registered with Caché and thereby regulated as part of the Caché security model. The user of a CSP application must be authenticated to Caché using one of the supported authentication mechanisms.

**current device**

*Caché System.* The device through which I/O commands are processed. When you log on, your current device is your principal device. This is usually the terminal or personal computer at which you logged on.

**current directory**

*General.* The directory in which you are currently working.

**cursor-based SQL**

*Caché SQL.* A type of [embedded SQL](#) query that opens a cursor to process the query. When your application needs to access multiple rows of data, you must use a cursor. A cursor acts like a pointer—it focuses on accessing and processing one row at a time, then moves from that row to the next in the sequence.

**cursor**

*Caché SQL.* A forward-moving iterator within multiple rows of data.

**custom storage**

*Caché Objects.* Custom storage allows you to determine the storage structure of objects by writing your own implementation of the storage interface methods. Generally, classes that use custom storage are not projected to SQL.

## 1.4 Terms Beginning with D

**DDL import utility**

*Caché Objects.* A utility included in the Caché Relational Utility Library to import DDL files into Caché. The utility automatically adds the corresponding definition to the Class Dictionary.

**DMNNET**

*Caché System.* A Caché process that handles incoming global requests from a network. It is the name that appears on the System Status (%SS) display or the Processes panel in System Operations Utilities to represent the network daemon.

**DSM**

*General.* InterSystems implementation of the M programming environment originally known as “Digital Standard M.”

**DTM NetBIOS**

*Caché System.* A network communication protocol that can be used to link to DTM databases.

## **DTM**

*General.* The InterSystems implementation of the M programming language designed to run on IBM-compatible PCs based on Intel 80386 and higher microprocessors.

## **Data Definition Language**

*Caché SQL.* Data Definition Language or DDL is a command based language used to create, define the structure of, and delete tables.

## **data location**

*Caché System.* The dataset in which a global resides. You define this in the data location column on the Namespaces panel of the Namespace/Network Configuration editor. The source of global sets and kills for a replicated global is the directory (or directory and system) mapped to the global's data location.

## **Data Source Name (DSN)**

*ODBC.* A data source name identifies a specific database on a specific server system.

## **data type class**

*Caché Objects.* A data type is a class with its DATATYPE class keyword set that supports the data type interface. This interface includes a number of operations designed for validation and SQL interoperability.

## **data type**

*Caché Objects.* The data type of an attribute determines its behavior, its validation requirements, and how it is projected to SQL, ActiveX, and Java.

## **database**

*General.* A CACHE.DAT file. It can contain code and data. Via global, package, and routine mappings, any given database can be used by multiple namespaces.

## **database encryption**

*Caché System.* The process by which a Caché database is stored on disk in an encrypted state. When Caché reads the data from the disk it is automatically decrypted at runtime so that its data can be available to its legitimate users. When the data is written to disk, it is encrypted. The data on disk is also referred to as being “at rest.”

## **database-encryption key**

*Caché System.* An AES key used to encrypt a Caché database.

## **database integrity**

*General.* The condition of a database being uncorrupted, either in its contents or its structure. The data in a database can sometimes become unreadable by degradation of its physical integrity. Internal pointers can become corrupted due to degradation of internal integrity.

## **dataset**

*General.* A logical entity that represents a directory name or directory and system name in Caché.

## **deep save**

*Caché Objects.* A deep save saves an object and all objects that it references, as described in class reference content for the %Persistent.%Save method.

**default dataset**

*Caché System.* The directory in which a namespace executes commands and receives any globals not independently mapped.

**default start routine**

*Caché System.* A default start routine is run automatically when a user logs in.

**Delphi**

*General.* A PASCAL-based GUI development tool that interfaces with Caché via ActiveX and ODBC.

**dependent relationship**

*Caché Objects.* A parent-child relationship that defines a child table for each row of the parent table.

**device accounts**

*Caché System.* Device accounts allow TELNET users to log in to specific routines and namespaces on your system remotely and locally. You may set up user accounts and/or device accounts using the Management Portal to facilitate better control over which users have access to specific routines, databases and namespaces.

**device**

*General.* A piece of hardware that is part of a computer system, such as a terminal, printer, disk drive, or magnetic tape drive.

**directory**

*General.* A name for a location on a disk where files can be stored. The Caché database files, CACHE.DAT and CACHE.EXT, reside in a directory. Only one Caché database file is permitted per directory.

**dismount, dismounted**

*Caché System.* To dismount a database is to disconnect it from an instance of Caché. When a database is dismounted, it must be explicitly mounted again for an instance to use it.

**display format**

*Caché Objects.* The display format of a property is the format used to display and input data.

**Distributed Cache Protocol (DCP)**

*Caché System.* The legacy internal networking system for Caché that was superseded by the Enterprise Cache Protocol (ECP). It was removed from Caché with version 2010.2.

**Distributed Data Processing (DDP)**

*Caché System.* A legacy message format protocol used for communication among instances of older InterSystems products and from Caché to those instances. Also known as DSM-DDP. It was removed from Caché with version 2010.2.

**distributed database**

*Caché System.* A database that is stored on multiple computers in a network. When you use Caché in a distributed database environment, ObjectScript routines that reside on one computer can access globals on other computers in the network, within the file protection limits established on those systems.

**dot syntax**

*Caché Objects.* Dot syntax allows you to get and set property values and to execute methods. It also allows you to access the properties and methods of referenced and embedded objects from the object referencing them.

**Dynamic Link Library (DLL)**

*General.* A library shared by various Microsoft Windows applications.

## 1.5 Terms Beginning with E

**embeddable class**

*Caché Objects.* Objects derived from an embeddable (serial) class can exist independently in memory, but, when stored to the database, exist only as data stored within a persistent object. See %SerialObject.

**embedded HTML**

*ObjectScript.* HTML that is directly embedded within ObjectScript methods or macro routines. Embedded HTML must be encased within &html<> statements.

**embedded object**

*Caché Objects.* An instance of an embeddable class. Embedded objects are separate objects in memory but can only be stored as data embedded within a persistent object.

**embedded SQL**

*Caché SQL.* [Embedded SQL](#) is SQL directly embedded within ObjectScript methods or macro routines. Embedded SQL must be encased within &sql() statements.

**empty**

*Caché Basic.* A value that indicates that no initial value has been assigned to a variable. Empty variables are 0 in a numeric context, or zero-length in a string context.

**empty string**

*Caché System.* A string that logically has no characters in it, often represented in text as " ". Some material may also refer to this as the “null string.”

Different languages differently represent the empty string, which determines if and how it occupies real space in computer memory. For example, in ObjectScript, " " does not occupy space in memory, while, in SQL, it does. Note also that a string with no characters in it is still a string and that, depending on the context, may be treated differently from the NULL (unassigned) value.

For example, in the following ObjectScript code:

**ObjectScript**

```
New A
New B
Set A = " "
Write A
Write B
```

the first **Write** statement succeeds (though this is not evident visually) and the second results in an <UNDEFINED> error.

**encapsulation**

*Caché Objects.* Encapsulation hides the internal details of a class by presenting a public interface that outlines all of the allowed interactions of the class without presenting any of the details used to perform those operations.

**endian**

*Caché System.* Whether a system is big-endian or little-endian refers to the order in which it stores the bytes of a multibyte element. Caché supports both big-endian and little-endian systems. It also includes a utility, **cvendian**, for converting databases from one form to the other. For details about cvendian, see the section on “[Using cvendian to Convert Between Big-endian and Little-endian Systems](#)” in *Caché Specialized System Tools and Utilities*.

**Enterprise Cache Protocol (ECP)**

*Caché System.* The internal networking system for Caché. ECP networking allows you to use Caché in a distributed database environment that contains a configurable number of nodes and a variety of hardware and software configurations.

**enumerated attribute**

*Caché Objects.* Enumerated attributes allow multiple choice values for a property value. The value of these attributes must be one of the predetermined choices.

**Ethernet address**

*General.* The identification of an Ethernet interface device for all interested computers. It is unique for each Ethernet interface on a network. Valid Ethernet addresses are nonzero, even, hexadecimal numbers of up to 12 digits.

**Ethernet link**

*General.* A high-speed coaxial cable that links multiple computers over a local area network.

**exclusive lock**

*General.* An exclusive lock prevents another process from viewing or editing the specified data.

If you use an exclusive lock when you open an object, it acquires an exclusive lock when the object is opened and releases it when the object is closed.

**explicit reference**

*Caché System.* Also called extended reference. A global reference that is made with a full definition of the system and namespace where the global resides.

**expression**

*Caché System.* A combination of keywords, operators, variables, and constants that yield a string, number, or object. An expression can perform a calculation, manipulate characters, or test data.

**expression method**

*Caché Objects.* An object method that may be placed in-line in the code generated by the class compiler.

**extended global reference**

*Caché System.* Also called explicit reference. A global reference that identifies the location of the global. The location can be specified as either a defined namespace or an implied namespace. Used when a Caché application needs to override the current namespace mapping for the global, which resolves to a different directory, or directory and system, than the one desired. For example, if the application is running in namespace ADMIN but needs to

refer to a global ^PARTS in the INVEN namespace, you can use the extended reference: ^["INVEN"]PARTS. If, on the other hand, a specific physical location is desired instead of the mapping currently in force for a particular namespace, you can use an implied namespace. To refer to global ^PARTS in the directory "inven" on the computer with the directory set name "production" on a UNIX® computer, you might make the extended reference: ^["^production^/usr/inven"]PARTS.

**extent index**

*Caché Objects.* An extent index maintains an index of all of the objects in an extent.

**extent query**

*Caché Objects.* A query that returns the ID of every object in the extent.

**extent**

*Caché Objects.* An extent spans the entire hierarchy tree of a specific class, called the root class. Caché SQL tables contain the entire extent of their corresponding class.

## 1.6 Terms Beginning with F

**F to CDL export utility**

*Caché SQL.* The F to CDL export utility converts data definitions in F.12 and later CDL files. These files can then be modified or loaded and compiled, creating classes from F constructs.

**F-DBMS**

*Caché SQL.* A legacy RDBMS (Relational Database Management System) predecessor of Caché; it is incompatible with Caché objects and Caché SQL.

**factory class**

*Caché Objects.* Objects instantiated from the Factory class in ActiveX or Java manage connections to Caché. They also create and open instances of other Caché classes and provide other administrative functionality.

**field**

*Caché Objects.* Another name for an SQL column.

**file stream**

*Caché Objects.* A file stream provides an interface to manipulate and store large chunks of text-based or binary data in an external file. The Caché stream interface can be used to manipulate file streams in ObjectScript, SQL, ActiveX, and Java.

**final class**

*Caché Objects.* A class that cannot be extended or subclassed.

**final method**

*Caché Objects.* A method that cannot be overridden.

**final property**

*Caché Objects.* A property that cannot be overridden.

**foreign key**

*Caché SQL.* A foreign key constrains a column in a table to point to another column in a table. The value supplied for the first column must exist in the second column.

## 1.7 Terms Beginning with G

**GSA file**

*Caché System.* A GSA file is a globals save file. The GSA file extension is not a requirement, but allows Caché and programmers to easily identify saved globals. Other than the file extension, there is no significance to a GSA file.

**Generic Buffered Interface (GBI)**

*Caché System.* A device I/O driver used by DTM to handle requests to and from external devices. GBIs are identified by a port number defined in the DTM.INI file.

**global database**

*Caché System.* The underlying logical and physical data storage structure of Caché, in which all data is stored in a system of multiply-subscripted arrays called "globals."

**global directory**

*Caché System.* A directory that contains a global database. It includes the database files and a list of all globals in the directory, with associated information.

**global**

*Caché System.* A multidimensional storage structure. [Globals](#) are implemented using balanced-tree technology within a Caché database.

**globally unique identifier (GUID)**

*Caché System.* A GUID is an identifier for an entity, such as an instance of a class, that is trusted to be unique for all instances of Caché, even across multiple systems. For example, if two independent instances of Caché use a common class definition that includes a GUID for each instance, then bringing the data together from these two instances will not result in any duplicate GUID values. Caché uses GUIDs as part of [object synchronization](#). For general information about using GUIDs, see the class documentation for %ExtentMgr.GUID and %Library.GlobalIdentifier.

## 1.8 Terms Beginning with H

**host name**

*General.* The name of the server system.

### **host variable**

*Caché SQL.* A variable that is linked, within an [embedded SQL](#) statement, to an application program variable.

## **1.9 Terms Beginning with I**

### **I/O translation**

*Caché System.* [National Language Support](#) facility consisting of a set of tables that transform between the character set of the computer and a particular device's character set. See also [Language Configuration](#).

### **ISM**

*General.* The InterSystems original implementation of the M language.

### **ITG file**

*Caché System.* An ASCII file that contains a database integrity report with an extension of .ITG for easy identification. An ITG file is created by a database integrity check.

### **identified by**

*Caché Objects.* A class is identified by another class when it is logically dependent on that class for its existence.

### **identifying relationship**

*Caché Objects.* An identifying relationship defines a relationship between two classes where one class is dependent upon the other for its existence.

### **identity**

*Caché Objects.* The identity, or ID, of an object uniquely identifies it within its extent.

### **idkey**

*Caché Objects.* An index that is used to designate the contents of the ID of an object. Any properties used in an idkey must remain static throughout the life of the object.

### **implicit global reference**

*Caché System.* See [Mapped Global Reference](#).

### **implied namespace**

*Caché System.* A namespace that Caché creates internally when you use a directory or directory and system name in an extended global reference.

### **in-memory value**

*Caché Objects.* The value of a property while it is in memory. For some types of properties, this can differ from its stored, or on-disk, value.



**include file**

*ObjectScript*. Files containing definitions that can be used in the preprocessor phase of ObjectScript source code compilation to expand macro source routines and determine whether optional lines of code should be included. They also can be used to include a common block of code in several routines, saving the overhead of calls to a common subroutine.

**incoming lock**

*Caché System*. A lock issued by a process on a remote client computer on an item on the local computer. It is also called a server lock, since the item being locked is located on the local computer, which is acting as a server. When you view locks, this lock appears in the Lock Table display with the system name of the remote computer that issued the LOCK request in the Owner column. The local server computer does not know which process on the remote client computer issued the LOCK nor does it track the number of locks on the item.

**index collation**

*Caché Objects*. Index collation specifies the data translation to use when storing data in an index.

**index**

*Caché Objects*. An index optimizes data retrieval by storing a sorted subset of the data for each object belonging to its class.

**inheritance**

*Caché Objects*. Inheritance passes the characteristics and members of a class to all of its subclasses. It allows you to group common aspects of multiple classes together in one superclass.

**install-dir**

*Caché System*. When generically referring to the Caché installation directory, the documentation uses the term *install-dir*. In its examples, the documentation uses C:\MyCache\. The section “[Default Caché Installation Directory](#)” describes where Caché is installed on all supported operating systems.

**instance method**

*Caché Objects*. A method that is invoked from a specific instance of a class and performs some action related to that instance.

**instance**

*Caché Objects*. An implementation of a class representing a specific entity. The terms “instance” and “object” can be used interchangeably.

**instantiate**

*Caché Objects*. To place an object instance into memory where a program can act upon it.

**intermediate source code**

*ObjectScript*. The standard 3GL (third generation language) ObjectScript source code available in Caché and all M implementations. Intermediate code is produced from macro source by the Caché compiler. At this level, all preprocessor syntax, including embedded SQL, has been resolved, and the routine contains only pure source code. It is possible to write ObjectScript routines at this level, but without the benefit of embedded SQL or other preprocessor syntax, such as macros.

**IP address**

*Caché System.* The numeric identifier for a computer according to the Internet Protocol (IP). Caché supports both IPv6 and IPv4 formats. For more information on Caché support for IPv6, see the section “[Use of IPv6 Addressing](#)” in the chapter “[Server Configuration Options](#)” in the *Caché Programming Orientation Guide*.

## 1.10 Terms Beginning with J

**JOBbed process**

*Caché System.* A background process created by issuing an ObjectScript JOB command at the Caché prompt or from within an application. On the relational server system, server masters, and the server processes they create, are examples of JOBbed processes.

**Java Database Connectivity (JDBC)**

*General.* Java Database Connectivity, or JDBC, provides a standard Java interface for relational data access.

**Java**

*General.* An object-oriented language originally created by Sun Microsystems.

**join**

*Caché SQL.* A request for information from a Caché relational database base table (via a query, form, or report) in which data must be retrieved from more than one table, necessitating a link between tables.

**journaling**

*Caché System.* A feature that, at the system manager’s option, causes Caché to keep a log of changes to all or selected globals in a journal file. If there is a system failure, these changes can be “rolled forward.” That is, an entire transaction can be reapplied to a database during a restore. See also Write Image Journaling.

## 1.11 Terms Beginning with K

**KDC**

*Caché System.* Key Distribution Center. As part of a Kerberos installation, this is the central Kerberos server that ensures the proper authentication of all parties. Specifically, the KDC is part of the trusted third-party Kerberos server that generates the keys that form the basis of ticket-granting tickets and service tickets. On Windows, the Key Distribution Center is part of the Windows Domain Controller (DC) and is sometimes called by that name. The commonality of the two initials is coincidental.

**Kerberos**

*Caché System.* A trusted third-party authentication system developed by Project Athena at MIT. It allows for the authentication of users or applications (collectively known as “principals”) by establishing a database of authentication information. This database is secured (and therefore trusted) and separate from any two parties performing authentication (which is why it is a “third-party” system). Kerberos is designed to be used on networks that are not necessarily secure, such as the Internet. It has been in use in large commercial and educational institutions since the late 1980’s.

**key (unique index)**

*Caché Objects.* A key is another name for a unique index.

**key (encryption)**

*Caché System.* A large number used with an encryption algorithm to encrypt or decrypt data.

**key-encryption key**

*Caché System.* With Caché database encryption, the second key involved in the process. The first key is used to encrypt the database and the key-encryption key — the second key — is used to encrypt (and therefore protect) the first key. When a database-encryption key is activated, it is decrypted with a key-encryption key and loaded into memory for use.

**keyword (class definition)**

*Caché Objects.* A keyword defines a specific characteristic within a class definition. Also known as a class keyword.

**keyword (system element)**

*Caché System.* A keyword may also refer to part of the Caché system, such as a function name or operator.

## 1.12 Terms Beginning with L

**language configuration**

*Caché System.* A set of four tables: character set, collation sequence, \$X/\$Y action, and pattern match; that define the device-independent aspects of National Language Support. Counterpart to the device-dependent National Language Support characteristic, I/O translation.

**legacy products**

*Caché System.* InterSystems products that are not currently in active development. These include Open M, DSM, DTM, ISM, and MSM.

**license**

*Caché System.* An agreement between InterSystems and its customer that defines the components of Caché software available to the customer and the number of users who can use each component. A customer must be licensed to run Caché. License information is distributed in a Product Activation Key and stored on your system in a file named cache.key.

**list**

*Caché Objects.* An ordered collection that uses slot numbers to access data. Each list is projected to SQL as a single list field.

**Local Area Network**

*General.* A network of limited geographic extent.

**Local Area Transport (LAT)**

*Caché System.* A legacy networking protocol originally supported for compatibility with DECnet (from Digital Equipment Corporation). It was removed from Caché with version 2010.2.

### **locale**

*Caché System.* The parameters that specify the user language, country, and any other, special variant preferences. A locale specifies user-visible conventions for the input, output, and processing of data, such as the representation of number and dates, and the names of days and months.

### **lock table**

*Caché System.* A Caché internal table where all LOCK commands issued by processes are stored. You can view this table by using the System Viewer.

### **log files**

*Caché System.* Files in the [system manager's directory](#) containing messages about system operations, errors and metrics. These include the console log (cconsole.log), System Monitor log (SystemMonitor.log), alerts log (alerts.log), initialization log (cboot.log), and journal history log (journal.log). For information about these log files, see “[Monitoring Log Files](#)” in the *Caché Monitoring Guide*.

### **log on**

*Caché System.* The act of signing on to the system. Users log on to Caché by entering their username and password when prompted.

### **logical format**

*Caché Objects.* The logical format of an object property is the format used in memory. All comparisons and calculations are performed on this format.

### **login role**

*Caché System.* Any role that becomes associated with a user by the action of authenticating to Caché (and not afterwards).

## **1.13 Terms Beginning with M**

### **MAP file**

*Caché System.* A DTM.MAP file describes the set of database resources to which each namespace will refer, and so, in effect, describes the set of network connections needed for accessing data.

### **MSM**

*Caché System.* The InterSystems implementation of the M programming language originally known as “Micro-netics Standard M.”

### **macro preprocessor**

*ObjectScript.* A part of the ObjectScript compiler that converts macro code into usable ObjectScript code.

### **macro source code**

*ObjectScript.* The highest, most flexible and permissive level of code at which routines and methods can be written. Macro source code permits the definition of macros and embedded SQL statements using a combination of ObjectScript syntax, special macro preprocessor commands, and ANSI-Standard SQL.

**map**

*Caché System.* In Caché, a definition that determines how data is stored when using Caché SQL Storage.

In Windows, a unit within a Caché database consisting of 400 2048-byte blocks, residing in a flat file.

In UNIX®, a unit within a Caché database consisting of 400 2048-byte blocks, residing in a single UNIX® file or raw partition.

**mapped global reference**

*Caché System.* A logical reference to a global that resides in a different directory, without using the extended reference syntax otherwise required to refer to a remote global. You can refer to the global as if it resides in the database's data location. That home directory can be on the same computer or any other computer on the network known to your Caché server. The system manager defines the actual location of a mapped global using the namespace/network configuration editor. Called an implicit reference or implicit global in versions prior to ISM 6.1.

**matching role**

*Caché System.* With a secured Caché application, a role that causes additional privileges to be granted. If a user holds a matching role, then, while using the application, that user is also granted whatever target roles are specified. Sometimes known as a “match role.”

**metadata class**

*Caché Objects.* A metadata class provides an interface to examine the data stored in object applications. See the `%Dictionary.ClassDefinition` class.

**metadata**

*Caché Objects.* Metadata describes data and how it is structured.

**method generator**

*Caché Objects.* [Method generators](#) are methods that generate runtime code based on the values of class parameters.

**method**

*Caché Objects.* An operation that can be invoked upon an object.

**mount, mounted**

*Caché System.* To mount a database is to explicitly connect it to an instance of Caché, thereby making its contents immediately available for use. A database in this state is described as mounted.

**multidimensional property**

*Caché Objects.* Multidimensional properties act like array nodes. Multidimensional attributes do not have any associated property methods, cannot be accessed using dot syntax, and are not projected to SQL, ActiveX, or Java.

**multidrop link**

*General.* Network hardware that connects more than two computers.

**multiple inheritance**

*Caché Objects.* Multiple inheritance allows a class to have more than one superclass. A class inherits characteristics and class members from each of its superclasses.

## 1.14 Terms Beginning with N

### **namespace**

*Caché System.* A logical entity that serves as a container for programs and data. Databases provides the physical storage for these, but the association of database objects logically is done through namespaces. For example, a namespace allows you to collect objects from multiple databases into a single frame of reference so that they appear as if they were all stored together. Users need only make a simple reference to a global no matter where it resides, while the namespace mapping specifies the global's database location.

For information on namespace names, see “[Configuring Namespaces](#)” in the chapter “Configuring Caché” in *Caché System Administration Guide*.

### **namespace map file**

*Caché System.* Open M [DTM] 6.3 for Windows or DOS uses a namespace \*.map file to define how directories are mapped to namespaces. Enter the name of the map file you want to use or click on the Browse button to search for the appropriate file.

### **National Language Support (NLS)**

*Caché System.* A Caché facility that helps you overcome differences among national languages that affect the way you enter, display, process, store, and transfer data. It consists of a set of tables that specify the internal character set, a collation sequence, pattern match, \$X/\$Y actions, and I/O translation. The system manager defines and loads these tables with the NLS utility. Users select tables for their processes and devices with the %NLS utility and the K and Y parameters to OPEN and USE commands.

### **network configuration**

*Caché System.* A description of the location of data within a network, and of the relationships among various components in the network. The combined entries in the Hardware Description Table, the Directory Set Location Table, and the DSM-DDR Volume Set Translation Table form this description. Although multiple configurations may be defined and stored in M/NET, only one at a time can be active.

### **network interface device**

*General.* The hardware that connects a computer to a network link.

### **network**

*General.* A collection of computers and connections, that allows users and programs on one computer to communicate with users and programs on other computers in the network.

### **node**

*Caché System.* One entry within a multidimensional array (global).

### **node**

*General.* One computer in a network or clustered system.

## 1.15 Terms Beginning with O

### **ODBC format**

*Caché Objects.* The ODBC format presents data with the formatting expected by ODBC.

### **ODBC Type**

*Caché Objects.* The ODBC type of a data type determines the ODBC data type used by properties based on the data type.

### **ODBC**

*General.* See Open Database Connectivity.

### **ODMG**

*General.* See Object Data Management Group.

### **OID**

*Caché Objects.* See object identifier.

### **OLE**

*General.* See ActiveX.

### **OREF**

*Caché Objects.* See object reference.

### **ObjInstance object**

*Caché Objects.* An ActiveX object that mirrors a specific object opened or created on the server.

### **object class**

*Caché Objects.* An object class represents a specific entity. It can have properties and be directly instantiated.

### **object code**

*Caché System.* The lowest level of code produced by the Caché compiler. This is the code that is actually interpreted and executed.

### **Object Data Management Group**

*General.* The Object Data Management Group, or ODMG, developed the original standard for object databases.

### **object identifier**

*Caché Objects.* An object identifier or OID uniquely identifies an object on disk within the entire database. The OID is valid for the life of an object and cannot be reused if the object is deleted.

### **object model**

*Caché Objects.* An object model describes the requirements of an application and the class hierarchy used in its development.

### **object reference**

*Caché Objects.* An object reference points to a specific object currently in memory. An object reference is only valid from the time an object is instantiated or opened until it is closed.

### **object-SQL projection**

*Caché Objects.* The object-SQL projection determines how Caché object functionality is projected to elements of Caché SQL.

### **object**

*Caché Objects.* A logical entity that encapsulates all of the data representing a specific item and the interface to manipulate that data.

### **ObjectScript query**

*Caché Objects.* An ObjectScript query uses ObjectScript code to query the database.

### **one-way outer join**

*Caché SQL.* A programmer-defined join that designates the first table specified in the join condition as the source table and includes all rows from the source table in the output table, even if there is no match in the second table. The source table pulls relevant information out of the second table but never sacrifices its own rows for lack of a match in the second table.

### **Open Database Connectivity (ODBC)**

*General.* The Microsoft Open Database Connectivity, or ODBC, provides a standard interface for data access on Windows.

### **Operating-System–based authentication**

*Caché System.* An authentication mechanism by which Caché accepts the identify of a user who has successfully authenticated to an operating system. The Caché user account must match the name of the operating system account. Also, once authenticated, the user's activities are restricted by the account's roles.

### **outgoing lock**

*Caché System.* A lock issued by a process on the local client computer on an item on a remote server computer. It is also called a client lock, since the local computer is the client. When you run the LOCKTAB utility from the Caché programmer prompt or the Locks utility in System Operations Utilities, this lock appears in the Lock Table display with an asterisk preceding the pid of the process issuing the lock. If incremental locks are issued on an item that already exists in the Lock Table, these requests are not sent across the network. Rather, the local computer keeps track of the number of incremental locks on outgoing locks.

### **override**

*Caché Objects.* By default, every class inherits the class members of its superclasses. However, unless a member is marked final, you can chose to modify the member so it acts differently in the subclass. This is called overriding the member.



## 1.16 Terms Beginning with P

### **package**

*Caché Objects.* A [package](#) contains a set of related classes.

### **parameter**

*Caché Objects.* A class parameter is a special constant set during application design time. Class parameters are available to all instances of a class and can be used for any purpose you wish. Every data type class has a set of class parameters that impose behavior on attributes based on that data type.

### **parent ID**

*Caché Objects.* In a Caché relational database, a field automatically created to define the Row ID in the dependent child table when a characteristic link is defined. The Parent ID acts like a designative reference from the child table to the parent table and has the same name as the parent table.

### **parent-child relationship**

*Caché Objects.* In a parent-child relationship, the existence of child object is dependent on the existence of a parent object. That is, a child object must be associated with a parent object and that child object is deleted when the parent object is deleted.

### **partition**

*Caché System.* Another name for the process private section of memory. Each process has its own partition for data private to that process. Also called the Caché partition or user partition. For UNIX®, see also Raw Disk Partition.

### **passing by reference**

*Caché System.* A way of passing the address, rather than the value, of an argument. This allows access to the actual variable so that the variable's actual value can be changed by the method, function, or routine to which it is passed.

### **passing by value**

A way of passing the value of an argument. This provides a copy of the variable. As a result, the variable's actual value cannot be changed by the method, function, or routine to which it is passed.

### **pattern match table**

*Caché System.* Internal table that tells Caché whether to treat characters as alphabetic, punctuation, numeric, or control characters.

### **permission**

*Caché System.* The ability to perform an activity on a resource. For database resources, the available permissions are Read and Write. For services, applications, or administrative actions, the available permission is Use.

### **persistent class**

*Caché Objects.* Objects of a persistent class can be stored in the database. Persistent classes inherit the persistent interface from the %Persistent class to manage data storage and retrieval.

### **persistent interface**

*Caché Objects.* The persistent interface is a set of methods used to store and retrieve objects.

### **point-to-point link**

*General.* A network link that connects two computers, one at each end.

### **polymorphism**

*Caché Objects.* Runtime references to overridden methods are dispatched based on the type of object it is. For example, if the Person class has a Student subclass with an overridden method Print(), Student objects will always use the Student Print() method even when they are opened as Person objects.

### **populate utility**

*Caché Objects.* The Populate Utility, implemented by the %Populate class, allows you to add dummy objects to your database for testing purposes.

### **port number**

*General.* A device number that identifies a particular network port. You must specify this number when connecting to a server from a client.

### **port**

*General.* A network software protocol/interface device combination. More than one port can use the same interface device.

### **primary key**

*Caché SQL.* A primary key is used by some systems to identify data.

### **primary persistent superclass**

*Caché Objects.* The primary persistent superclass of a class determines the persistent behavior of that class. By default, the primary persistent superclass is the leftmost persistent superclass in the list of superclasses. Typically all data associated with classes with same primary persistent superclass is stored together.

### **primary volume**

*Caché System.* The first or only volume in a volume set.

### **principal device**

*Caché System.* The input/output device associated with your process, usually your terminal or computer keyboard and monitor. For jobbed processes, you can assign the principal device in the JOB command or set it as the principal device of the parent process in the System Configuration editor. If you do not set the device in one of these ways then the default device for jobbed processes is the null device.

UNIX®: For interactive users, the terminal at which the user enters Caché. It is the same as the principal input device UNIX® has assigned to that user. For jobbed processes, the principal device can be assigned by the JOB command, and is /dev/null by default.

### **priority setting**

*Caché System.* In UNIX® systems, priority is the system-level definition of the order in which jobs are assigned system resources and responses.

For Windows systems, “Priority” has no meaning for individual processes, and the default priority for interactive jobs is 0.

**priority**

*Caché System.* The importance of a batch job or system daemon. As a way to balance system resources, you can adjust the priorities of jobs or daemons in order to achieve the best overall performance for your system.

**private**

*Caché Objects.* A private class member can only be accessed by methods belonging to that class. Both methods and properties can be private.

*Caché System.* In Caché Basic, a private variable is only visible in the script in which it is created.

**privilege**

*Caché System.* A specification of the ability to perform a particular action on a particular resource, such as being able to read the DocBook database. Only roles can hold privileges.

**privileged routine application**

*Caché System.* A set of one or more Caché routines that are grouped together for security purposes. Such a group is treated as a single application and is represented by a single resource.

**procedure**

*Caché System.* A named sequence of statements executed as a unit.

**process-private section**

*Caché System.* The area of memory used solely by a particular process.

**process**

*Caché System.* An entity scheduled by the system software, which provides a context in which an image executes. Within Caché, a process is the context in which server-based code executes.

**Product Activation Key**

*Caché System.* A paper key that arrives with your software distribution on which is printed an encoded version of your Caché license. You must enter this information into a file called CACHE.KEY in order to activate the license.

**programmer mode**

*Caché System.* The mode in which all program development activity takes place. In programmer mode, you initiate programs from the Caché prompt, and the Caché prompt reappears at the conclusion of every program you run. Programmer mode encompasses the Caché environment and all programs that can be called from it, including the Caché SQL development environment and environment. In programmer mode, you can create applications that users subsequently run in [application mode](#).

**project**

*Caché System.* A user-defined list of related source documents (such as routines or class definitions). Projects are used by Studio to help organize application development.

### **property collation**

*Caché Objects.* Property collation overrides the default translation provided by an attribute's data type with a collation specified in the attribute definition.

### **property method**

*Caché Objects.* A method generated to control behavior of a property. The name of the property is concatenated with the behavior methods from its data type and property behavior classes at compile time to create a full set of property methods for the property.

### **property**

*Caché Objects.* A data element representing a specific characteristic associated with a class. Each object of that class contains a literal value for the property or a reference to another object representing the data. Together, a class's properties represent its state.

### **public**

*Caché Objects.* A public class member can be seen and used by any class.

*Caché System.* In Caché Basic, a variable is declared with the Public statement and which is visible to all procedures in all modules of all applications.

## **1.17 Terms Beginning with Q**

### **query interface**

*Caché Objects.* The Caché query interface provides a common mechanism for preparing, executing, and processing queries regardless of the type or language of the queries.

### **query**

*General.* A request for data that meets the specified criteria. Caché supports predefined queries in SQL or ObjectScript (defined in classes) or ad-hoc queries (from embedded SQL or ODBC).

## **1.18 Terms Beginning with R**

### **RECEIVE**

*Caché System.* A Caché system process that broadcasts Caché network configuration information to all known remote computers.

### **RSA file**

*Caché System.* An RSA file is a (pre-XML) routine save file. The RSA file extension is not a requirement, but allows Caché and programmers to easily identify saved routines. Other than the file extension, there is no significance to an RSA file.

### **range indicator**

*Caché System.* A range indicator (:) allows you to indicate a range of globals in one entry. For example A:Cost would include all globals from ^Aardvark to ^Cost, but would not include ^CrossReference.

**raw disk partition**

*Caché System.* A UNIX® operating system division of a disk, that is not associated with a UNIX® file system. When a Caché database resides in a raw disk partition, it does not have UNIX® file name.

**reconfiguration**

*Caché System.* In Caché networking, the alteration of either network hardware or remote system, dataset or namespace definitions.

**recovery**

*Caché System.* A method of recovering from a loss of data integrity after a system failure.

**redirection**

*Caché System.* A way to make the location of a global invisible to an application. Caché supports two types of redirection: namespace definition and replication.

**reference**

*Caché Objects.* A reference points to another object or table, creating a one-way relationship between a property or field and the referenced object or table.

**registered class**

*Caché Objects.* A registered class is derived from the %RegisteredObject class. Caché automatically manages object references for registered classes and supports polymorphism.

**relational database**

*General.* A collection of related data that is organized according to the relational model.

**relationship**

*Caché Objects.* A two-way reference between properties of two different classes.

**relative dot syntax**

*ObjectScript.* Relative dot syntax (..) is used to reference a property or invoke a method of the current class.

**remote computer**

*Caché System.* Any computer that is linked to the local computer via a network link.

**replicated global**

*Caché System.* A global whose namespace mapping defines one or more locations for duplicate copies in addition to the primary data location. Any SET or KILL operations on the original copy of a replicated global are performed on all copies of that global. Any SET or KILL operations performed on a copy of a replicated global only affect the copy.

**required field**

*Caché Objects.* A field in a table that must contain a valid non-null value before the row can be filed. See Required Property.

**required property**

*Caché Objects.* A required property must be assigned a value before its object can be saved.

## resource

*Caché System.* The smallest granular unit that can be protected with Caché security. A resource represents one or more assets, which can be databases, access to applications, or other elements. Generally, resources can only include homogeneous sets of sets of assets.

## result set

*Caché Objects.* A result set contains the data returned by a query. It can be processed using the Query Interface. See %ResultSet.

## role

*Caché System.* An entity within Caché security that can receive the ability to both perform actions and view or alter data. These abilities are granted to a role in the form of privileges. When a user becomes a member of a role, it receives the role's privileges.

## role-assignment mechanism

*Caché System.* A means by which a user is given various [roles](#) for using Caché. Role assignment can occur using a native Caché mechanism, LDAP, or delegated (that is, user-defined) code. It is part of the [authorization](#) process.

## roll back

*Caché System.* A process by which an incomplete transaction can be removed from your Caché database. Caché removes such changes as part of its recovery procedure. See Journaling, Transaction.

## roll forward

*Caché System.* The process of reapplying journaled changes to the database, in the event that a system failure occurs and a database is recreated from backups. See Journaling.

## root class

*Caché Objects.* The root class of an extent is the top class in its hierarchy. Every class is the root class of its own extent. However, the term “root class” generally refers to the primary persistent superclass of a class hierarchy, especially in the context of data storage.

## routine buffers

*Caché System.* The amount of virtual memory that a routine is allowed to use.

## routine

*Caché System.* In Caché, an ObjectScript or Basic program. Routines are stored in a Caché database.

## Row ID

*Caché Objects.* The Row ID uniquely identifies a row in a table. If the table is projected from a Caché class, the Row ID is the ID of the equivalent object (either generated automatically or specified by an ID Key).

## row specification

*Caché Objects.* The row specification of a class query provides information on the fields returned by that query.

**row**

*Caché SQL.* A group of related field values that describes an entity in the domain of a relational table. For example, in a Customers table, a row describes a single customer. Also called a “record” in traditional data processing terminology, or a “tuple” in relational database terminology.

## 1.19 Terms Beginning with S

**SOAP**

*General.* A protocol specification for the implementation of web services. It uses XML as its message format, and usually relies on other application layer protocols — such as Remote Procedure Call (RPC) and HTTP — for message negotiation and transmission. **SOAP** was formerly an acronym that stood for “Simple Object Access Protocol,” but it is no longer an acronym — it is simply the name of the protocol.

**SQL computed field**

*Caché Objects.* An SQL computed field is calculated on demand and based on values stored in other fields.

**SQL mapping**

*Caché Objects.* SQL mapping is used to map existing data structures to classes and tables.

**SQL query**

*Caché SQL.* An SQL query uses SQL to locate and retrieve information stored in the database.

**SQL reserved words**

*Caché SQL.* **SQL reserved words** have special meaning in Caché SQL and cannot be used as table, field, query, view, stored procedure, or index names. SQL reserved words can be used as class and property names if alternate table and field names are defined.

**SQL**

*General.* Structured Query Language. SQL is the ANSI-standard fourth generation programming language designed specifically for accessing and maintaining relational data bases.

**SQLCODE**

*Caché SQL.* A local variable that holds the status of executed SQL statements. If an error occurred during execution, the SQLCODE contains the number of the error.

**schema evolution**

*Caché Objects.* Schema evolution allows you to add new class members to a class without losing the ability to access data stored using the old class definition.

**search user**

*Caché System.* The user with which Caché connects to an LDAP server and whose privileges allow for searching the LDAP database. Its value is specified in the **LDAP username to use for searches** field on the **LDAP Options** page (**System Administration > Security > System Security > LDAP Options**).

## **secondary volume**

*Caché System.* Any volume in a volume set other than the primary volume. There can be from zero through seven secondary volumes in a Caché database, for a total of eight volumes.

## **Security Advisor**

*Caché System.* A diagnostic tool within the Management Portal that provides feedback on those areas in which a Caché instance's setup differs from that suggested by typical security standards.

## **security domain**

*Caché System.* A logical grouping of machines that roughly corresponds to a Kerberos realm or Windows domain.

## **selectivity**

*Caché SQL.* The selectivity of a property specifies the approximate frequency of specific values within the entire distribution of values.

## **sequential file**

*General.* A file whose data is stored in the order in which it is entered.

## **server lock**

*Caché System.* See Incoming Lock.

## **server master**

*Caché System.* A component of the server software that listens for connection attempts from clients connected via TCP, and spawns server processes to service those connections. Each server master is a Caché process.

## **server process**

*Caché System.* A JOBbed process on the server that services a single client connection. A Server Master creates a server process for this purpose.

## **service**

*Caché System.* An entity within Caché security that regulates access to Caché through an existing pathway, for example, Telnet, ODBC, or CSP. An application sits on top of a service when connecting to Caché.

## **shadowing**

*Caché System.* Shadowing enables a secondary computer to maintain a copy or image of selected databases located on a primary machine. Shadowing requires a source database server (primary system) and a destination shadow server (secondary system)

## **shallow save**

*Caché Objects.* A shallow save saves an object but not the objects that it references. For more details, see the class reference content for the %Persistent.%Save method.

## **shared lock**

*Caché Objects.* A shared lock holds a shared lock on an object while it is loaded from the database and an exclusive lock while it is being saved if it spans more than one node or if it is being updated. No lock is obtained for an object in a single node being saved for the first time.



**shared retained lock**

*Caché Objects.* A shared retained lock holds a shared lock on an object from the time it is loaded from the database until it is closed and an exclusive lock while it is being saved if it spans more than one node or if it is being updated. No lock is obtained for an object in a single node being saved for the first time.

**sign on**

*Caché System.* The act of entering Caché in order to use Caché or a Caché application. A user must log on to the operating system before signing on to Caché.

**special variable**

*Caché System.* One of many variables that Caché maintains. Application developers can examine special variables to get information about the system.

**state**

*Caché Objects.* The state of an object is comprised of the current value of each of its properties.

**status code**

*Caché Objects.* A status code is returned by some methods and indicates whether the method was successfully executed. See %Status.

**storage definition**

*Caché Objects.* A storage definition consists of a storage class and any defined SQL mappings. Together, these elements determine how data is stored.

**storage interface**

*Caché Objects.* A set of methods that must be implemented when using custom storage or writing your own storage class.

**storage strategy**

*Caché Objects.* The storage strategy used by a class evaluates to a storage definition at compile time that determines how data is stored.

**stored procedure**

*Caché SQL.* Stored procedures allow you to execute a query or class method from ODBC or JDBC.

**stream interface**

*Caché Objects.* The Caché stream interface is used to manipulate streams in ObjectScript, SQL, ActiveX, and Java.

**stream**

*Caché Objects.* A stream provides an interface to manipulate and store large chunks of data. The Caché stream interface can be used to manipulate streams in ObjectScript, SQL, ActiveX, and Java.

**subclass**

*Caché Objects.* A subclass inherits class members and characteristics from another class, its superclass. A subclass typically must pass the “Is A” test with its superclasses—a subclass *Is a* superclass.

## **superclass**

*Caché Objects.* A class inherits the specifications of its superclasses, including properties, methods, class parameters, applicable class keywords, and the parameters and keywords of the inherited properties and inherited methods.

## **superserver**

*Caché System.* The server that listens on a specified port for incoming connections to Caché and dispatches them to the appropriate subsystem. The default superserver port is 1972; if 1972 is not available, the superserver listens on port 56772 or to the next available number higher than 56772. To set the superserver port number, use the **Superserver Port Number** field of the **Memory and Startup** page of the Management Portal (**System Administration** > **Configuration** > **System Configuration** > **Memory and Startup**).

## **swizzling**

*Caché Objects.* The process of automatically pulling embedded and persistent objects into memory as soon as they are referenced. Also known as “lazy loading.”

## **SysList class**

*Caché Objects.* SysList is a class in the Caché Object Server for ActiveX used to manipulate lists including properties using and methods returning the %List data type.

## **system class**

*Caché Objects.* The classes that provide built-in functionality to Caché.

## **system configuration**

*Caché System.* A definition of the system resources that Caché uses at startup. You define system configurations with the Management Portal.

## **system manager's directory**

*Caché System.* The directory where the Caché database resides that contains Caché system globals, system routines and routines for Caché manager and % utilities. It is the subdirectory MGR in your Caché installation directory.

## **system name**

*Caché System.* The name assigned to a node in a network. It must be unique across the network. Also called host name or computer name. Used in the Namespace/Network Configuration Editor to identify a computer in a network configuration. Called directory set in the MNET utility.

## **system processes**

*Caché System.* Windows - On Windows, you cannot adjust the priorities of the processes.

UNIX® - On UNIX®, priority is controlled by nice values. By raising the nice value for a process, you give it a lower priority; by lowering the nice value, you give the process a higher priority.

## 1.20 Terms Beginning with T

### TCP/IP

*General.* Transmission Control Protocol/Internet Protocol, one of the communications protocols that can govern the connection between the Relational Client and Relational Server. Also referred to as TCP.

### table

*Caché SQL.* A data structure composed of rows representing specific entities and columns representing a specific data point about each entity.

### target role

*Caché System.* Within a secured Caché application, a role that is granted by an application to a user who is already the member of some other role (called a matching role). If a user holds a matching role, then, while using the application, the user can be granted one or more additional target roles.

### target user

*Caché System.* A user who is attempting to authenticate to an LDAP server. Caché attempts to locate this user in the LDAP database by using the value supplied for the **LDAP Unique search attribute** field on the **LDAP Options** page (**System Administration > Security > System Security > LDAP Options**).

### transaction

*General.* A logical unit of work. Application developers can define transactions using SQL or ObjectScript commands. Caché logs updates to globals in a transaction in the journal file. If a transaction is incomplete, it can be rolled back.

### transient property

*Caché Objects.* A transient property is stored in memory but not on disk.

### translation methods

*Caché Objects.* The translation methods are used to convert values between ODBC, display, logical, and storage formats.

### trigger

*Caché SQL.* A sequence of actions defined by the developer to execute at various points during a Caché SQL application or an object application using Caché SQL Storage. Triggers are database actions initiated by INSERT, UPDATE, or DELETE actions performed on a table. Triggers help maintain integrity constraints and other data dependencies.

## 1.21 Terms Beginning with U

### UIC

*Caché System.* User Identification Code. The UIC of a database determines who can have access to globals in that database. On UNIX®, a UIC consists of a groupid, followed by a comma (,) and a userid; the system manager assigns UICs to users at the UNIX® level, and uses the Caché MSU utility to assign a UIC to each database.

### **unauthenticated access**

*Caché System.* Access to Caché that is not based on any verification of user identity.

### **Unicode**

*Caché System.* A Caché collation designed to handle a 16-bit character set. This allows for 64K possible characters as opposed to ASCII which allows only 256 characters. This collation allows for smooth transition to foreign character sets containing more than 256 characters.

### **Unified Data Architecture**

*Caché Objects.* The Unified Data Architecture permits Caché to automatically maintain synchronized object and relational data structures for all data, allowing each user to choose their own access path to the data.

### **unique index**

*Caché Objects.* A unique index specifies that each object has a unique value for the attribute or combination of attributes in the index.

### **unmounted**

*Caché System.* An unmounted database is available for use by an instance of Caché but is not in active use. A reference to any of the contents of an unmounted database implicitly mounts it.

### **user account**

*Caché System.* The entity that allows individual users to log in to specific routines and namespaces on your system remotely and locally. You set up user accounts and/or device accounts using the System Operations Utility User Accts to facilitate better control over which users have access to specific routines, databases and namespaces.

### **user process**

*Caché System.* A Caché interactive process that is associated with a specific terminal at which the user has signed on to Caché.

## **1.22 Terms Beginning with V**

### **valid subscript ranges**

*Caché System.* Subscript ranges are valid when all subscripts in the range are valid. Examples of valid subscript ranges would be: (1):(10) or ("a")("P"). The first subscript in the range is inclusive, while the second is exclusive. Thus (1):(10) includes the subscript value 1 but does not include the value 10. This allows Caché to define an upper boundary on the range.

### **variable**

*Caché System.* A symbolic name that is used to reference a data value. Variables can be local or global. Local variables reside in the local symbol table associated with a given partition. Global variables reside on disk.

### **view**

*Caché SQL.* A [view](#) presents data from one or more tables that meets a specified criteria.

**virtual field**

*Caché SQL.* A field that does not correspond directly to a single stored value but instead is composed of several stored values. For example, the Row ID field for a child table is sometimes composed of two stored values—the Row ID of the parent and a subscript corresponding to a particular child row.

**virtual table**

*Caché SQL.* A named table derived from one or more base tables that is not directly represented in physical storage. Views and query output are examples of virtual tables.

**Visual Basic**

*General.* A Microsoft GUI development environment that interfaces with Caché via ActiveX and Visual Caché components.

**volume set**

*Caché System.* A collection of one or more volumes that collectively contain all of the maps comprising a Caché database. A Caché directory or dataset is the same as a volume set. A volume set always consists of one primary volume and from zero through seven secondary volumes, for a maximum of eight volumes.

## 1.23 Terms Beginning with W

**watchpoint**

*Caché System.* Variable that you identify in a [ZBREAK](#) command. When its value is changed via a SET or KILL command, you can cause the interruption of routine execution and/or the execution of debugging actions you define within the ZBREAK command.

**wide-area network**

*General.* A network of large geographic extent, typically greater than one kilometer.

**write image journaling**

*Caché System.* A way to update a Caché database that minimizes the possibility of degradation of the internal integrity of the database, by retaining a copy of all modifications that the Write Daemon has not yet actually written into the database.

## 1.24 Terms Beginning with X

**XData**

*General.* A class member that holds a named block of well-formed XML for use after compilation. While an XData block may contain any structured information, it is typically used with Ensemble or Zen classes.

**XML**

*General.* [eXtensible Markup Language](#). A standard and set of technologies for defining and working with customized markup languages.

## 1.25 Terms Beginning with Z

### **Zen**

*Caché System.* An application framework from InterSystems for creating Web applications from pre-built object components. These components automatically create standard HTML and JavaScript needed to render the applications; they also provide a common object model that is shared between the user's browser and the application logic running on the server. Zen is based on Caché Server Page (CSP) and Caché objects.