InterSystems Glossary of Terms

Version 2024.2
2024-09-05

InterSystems Glossary of Terms
PDF generated on 2024-09-05
InterSystems IRIS® Version 2024.2
Copyright © 2024 InterSystems Corporation
All rights reserved.

For Support questions about any InterSystems products, contact:

InterSystems Worldwide Response Center (WRC)

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

Table of Contents

1

Complete Glossary of Terms

The following is a list of terminology relevant to InterSystems IRIS® data platform.

1.1 Terms Beginning with Symbols and A

$X/$Y action table

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 InterSystems IRIS. 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.

abstract class

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

abstract persistent class

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

access mode

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 InterSystems IRIS on the server machine); client-server (where the user connects to InterSystems IRIS from a separate executable, usually on a separate machine); and web (where the user interacts with InterSystems IRIS through a web application that uses the Web Gateway).

activate

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

added role

> *System*. Any role that becomes associated with a user after that user authenticates to InterSystems IRIS.

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

> *System*. The mode in which InterSystems IRIS 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 InterSystems IRIS 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.

array

> *Objects*. A collection that uses key-value pairs to access data. Arrays are projected to SQL as child tables.

ASCII character set

> *System*. For InterSystems products, ASCII refers to extended, 8-bit character sets, rather than the more limited, 7-bit character set. By default, an InterSystems IRIS instance uses the character set from the machine's locale, as specified in Supported Languages for this release; if there is no support for a machine's default language, InterSystems IRIS uses the ISO-8859-1 character set, commonly known as Latin-1.

asset

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

atomic lock

> *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

> *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

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

authentication

> *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

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

authorization

> *System*. The process of determining what an authenticated user can do on an InterSystems IRIS 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 InterSystems IRIS.

1.2 Terms Beginning with B

binary stream

> *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 InterSystems IRIS stream interface can be used to manipulate binary streams in ObjectScript, SQL, and Java.

bind

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

BLOB

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

breakpoint

> *ObjectScript*. Location in an InterSystems IRIS routine that you specify with the ZBREAK command. When execution reaches that line, InterSystems IRIS 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, InterSystems IRIS caches globals in global buffers, and routines in routine buffers.

1.3 Terms Beginning with C

CLASSPATH

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

calculated property

> *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

> *Objects*. A method that directly invokes an InterSystems IRIS routine.

callin interface

> *System*. InterSystems IRIS 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

> *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. InterSystems IRIS 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 $NUMBER.

> For information on the canonical form of globals and the related naked global references, see Naked Global References or the $NAME or $ZREFERENCE reference pages.

> 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 ..\iris.wij is not in canonical form, while C:\InterSystems\MyIRIS\IRIS.WIJ is (if there is an instance of InterSystems IRIS called MyIRIS and installed in the standard location).

> For UNIX®, an example of the canonical form of a device name is /dev/tty.

cascading dot syntax

> *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

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

character stream

> *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 InterSystems IRIS stream interface can be used to manipulate character streams in ObjectScript, SQL, and Java.

child table

> *InterSystems 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

Objects. A set of two properties in different classes where each property is a reference to the other class. InterSystems IRIS 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

Objects. The class compiler compiles InterSystems IRIS classes. You can use the class compiler from your IDE or by using one of the compile calls in the InterSystems IRIS object utility library.

Class Definition Language

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

class descriptor

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

class dictionary

Objects. The class dictionary holds the current class definition of all compiled classes. The class dictionary is also used by InterSystems SQL to determine valid table definitions.

class hierarchy

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

class member

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

class method

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

class

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 indexes.

InterSystems IRIS 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

Objects. The client data type is used to specify information needed to project data to clients via an InterSystems IRIS 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 SQL and clients such as Java.

client lock

System. See outgoing lock.

client

> *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

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

code method

> *Objects*. A method that executes ObjectScript.

collation rule

> *System*. An algorithm that determines the order in which InterSystems IRIS retrieves global nodes. InterSystems IRIS 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

> *System*. Specifies the order in which InterSystems IRIS 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

> *InterSystems SQL*. Collation specifies how InterSystems 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. Indexes use the collation type of the properties they index unless an index collation type is explicitly specified.
>
> InterSystems 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.

collection

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

column

> *InterSystems 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

> *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 InterSystems IRIS handles compiling classes with objects in memory.

compilation

> *Objects*. The process which transforms InterSystems IRIS classes and routines into executable code.

compiler directive

> *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

> *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

> *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

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

concurrency

> *Objects*. Concurrency provides a mechanism for controlling data integrity through locking. InterSystems IRIS 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, InterSystems IRIS uses atomic locking for all objects.

concurrent backup

> *System*. A backup performed without stopping other database activity. InterSystems IRIS 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

> *System*. An InterSystems IRIS configuration describes InterSystems IRIS resources at startup. You define a configuration in the Management Portal. You can create more than one configuration, although only one can be current at a time. InterSystems IRIS uses the current configuration at startup.

connection

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

connection security level

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

System. The means by which users establish their connection with InterSystems IRIS (such as the Terminal or Java).

conversion code

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

CPF

System. Short for Configuration Parameter File, a CPF is a file with a .cpf extension that defines a configuration. See Configuration.

credentials cache

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.

current device

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

InterSystems 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

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

custom storage

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

database

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

database cache

System. System memory (RAM) used to cache data retrieved from databases, so that repeated instances of the same query can retrieve results from memory rather than storage, providing a very significant performance benefit. For best performance, the database cache should be at least as big as the application workload's working set.

database encryption

System. The process by which an InterSystems IRIS database is stored on disk in an encrypted state. When Inter-Systems IRIS 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

System. An AES key used to encrypt an InterSystems IRIS 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.

Data Definition Language

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

data location

System. The dataset in which a global resides. 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.

dataset

General. A logical entity that represents a directory name or directory and system name in InterSystems IRIS.

Data Source Name (DSN)

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

data type

Objects. The data type of an attribute determines its behavior, its validation requirements, and how it is projected to SQL and clients such as Java.

data type class

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.

DDL import utility

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

deep save

> *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

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

default start routine

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

dependent relationship

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

device accounts

> *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.

directory

> *General*. A name for a location on a disk where files can be stored. The InterSystems IRIS database file, IRIS.DAT, resides in a directory. Only one InterSystems IRIS database file is permitted per directory.

dismount, dismounted

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

display format

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

distributed database

> *System*. A database that is stored on multiple computers in a network. When you use InterSystems IRIS 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.

DMNNET

> *System*. An InterSystems IRIS 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.

dot syntax

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

1.5 Terms Beginning with E

embeddable class

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

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

InterSystems SQL. Embedded SQL is SQL directly embedded within ObjectScript methods or macro routines. Embedded SQL must be encased within &sql() statements.

empty string

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

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

System. Whether a system is big-endian or little-endian refers to the order in which it stores the bytes of a multibyte element. InterSystems IRIS 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.

Enterprise Cache Protocol (ECP)

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

enumerated attribute

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

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

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 method

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

extended global reference

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 an InterSystems IRIS 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

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

extent query

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

extent

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

1.6 Terms Beginning with F

factory class

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

field

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

file stream

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

final class

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

final method

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

final property

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

foreign key

> *InterSystems 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.

foundation

> *Health care.* In InterSystems IRIS for Health™ and HealthShare®, a namespace that is enabled for health care interoperability.

1.7 Terms Beginning with G

global

> *System.* A multidimensional storage structure. Globals are implemented using balanced-tree technology within an InterSystems IRIS database.

globals database

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

global directory

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

globally unique identifier (GUID)

> *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 InterSystems IRIS, even across multiple systems. For example, if two independent instances of InterSystems IRIS 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.InterSystems IRIS uses GUIDs as part of object synchronization. For general information about using GUIDs, see the class documentation for %ExtentMgr.GUID and %Library.GlobalIdentifier.

GSA file

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

1.8 Terms Beginning with H

host name

> *General*. The name of the server system.

host variable

> *InterSystems SQL*. A variable that is linked, within an embedded SQL statement, to an application program variable.

1.9 Terms Beginning with I

identified by

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

identifying relationship

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

identity

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

idkey

> *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

> *System*. See Mapped Global Reference.

implied namespace

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

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

> *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

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

index collation

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

inheritance

> *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.

in-memory value

> *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.

install-dir

> *System*. When generically referring to the InterSystems IRIS installation directory, the documentation uses the term *install-dir*. In its examples, the documentation uses C:\MyIRIS\. The section Default Installation Directory describes where InterSystems IRIS is installed on all supported operating systems.

instance

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

Instance Authentication

> *System*. The native InterSystems authentication system: a user is prompted for a password, a hash of the supplied password is passed to the InterSystems IRIS server, and that hash is compared to the hash of the existing password stored in the server. If the two values are identical, InterSystems IRIS then grants authorization to those resources for which the user has privileges.
>
> This mechanism is listed in the Management Portal as **Password** authentication.

instance method

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

instantiate

> *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 InterSystems IRIS. Intermediate code is produced from macro source by the InterSystems IRIS compiler. At this level, all pre-processor 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.

InterSystems IRIS launcher

> *System*. The icon that appears on the taskbar when you start InterSystems IRIS on a Windows system. From this icon, you can configure and manage your InterSystems IRIS systems as well as create and manage classes and routines.

InterSystems IRIS database

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

InterSystems IRIS Object Server for Java

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

InterSystems IRIS object utility library

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

InterSystems IRIS relational utility library

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

InterSystems IRIS server

> *System*. Facility that allows you to use a distributed InterSystems IRIS database in a networked system.

InterSystems IRIS storage

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

InterSystems SQL

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

InterSystems SQL server

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

InterSystems SQL storage

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

I/O translation

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.

IP address

InterSystems IRIS System. The numeric identifier for a computer according to the Internet Protocol (IP). InterSystems IRIS supports both IPv6 and IPv4 formats. For more information on InterSystems IRIS support for IPv6, see Use of IPv6 Addressing.

IRIS.DAT

System. The primary volume in an InterSystems IRIS database. It contains InterSystems IRIS globals and routines.

ITG file

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.

1.10 Terms Beginning with J

Java

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

Java Database Connectivity (JDBC)

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

JOBbed process

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

join

InterSystems SQL. A request for information from an InterSystems IRIS 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

> *System*. A feature that, at the system manager's option, causes InterSystems IRIS 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

> *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

> *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)

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

key (encryption)

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

key-encryption key

> *System*. With InterSystems IRIS 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)

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

keyword (system element)

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

1.12 Terms Beginning with L

language configuration

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.

license

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

list

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

locale

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

System. An InterSystems IRIS internal table where all LOCK commands issued by processes are stored. You can view this table by using the System Viewer.

log files

System. Files in the system manager's directory containing messages about system operations, errors and metrics. These include the messages log (messages.log), System Monitor log (SystemMonitor.log), alerts log (alerts.log), initialization log (iboot.log), and journal history log (journal.log). For information about these log files, see Monitoring Log Files.

logical format

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

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

1.13 Terms Beginning with 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

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

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

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

mapped global reference

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 InterSystems IRIS server. The system manager defines the actual location of a mapped global using the namespace/network configuration editor.

matching role

System. With a secured InterSystems IRIS 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

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

metadata

Objects. Metadata describes data and how it is structured.

method generator

Objects. Method generators are methods that generate runtime code based on the values of class parameters.

method

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

mount, mounted

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

multidimensional property

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 or Java.

multidrop link

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

multiple inheritance

>*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

>*System*. A logical entity that provides access to data and code physically stored in databases. References to database objects are made logically through namespaces rather than to physical database locations. Namespace mappings specify the database locations of globals so users and code need only make a simple reference to a global in a namespace, without having to be concerned with the physical location of the data it contains. For example, namespace mappings allow you to collect objects from multiple databases into a single namespace, which serves as a single frame of reference, as if they were all stored together.

>For information on namespace names, see Configuring Namespaces.

National Language Support (NLS)

>*System*. An InterSystems IRIS 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 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

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

node

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

1.15 Terms Beginning with O

object

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

object class

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

object code

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

object identifier

> *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

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

object reference

> *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.

ObjectScript

> *System*. One of the programming languages supported by InterSystems IRIS.

ObjectScript query

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

object-SQL projection

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

ODBC

> *General*. See Open Database Connectivity.

ODBC format

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

ODBC Type

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

OID

> *Objects*. See object identifier.

one-way outer join

> *InterSystems 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

> *System*. An authentication mechanism by which InterSystems IRIS accepts the identify of a user who has successfully authenticated to an operating system. The InterSystems IRIS 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.

OREF

> *Objects*. See object reference.

outgoing lock

> *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 Terminal 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

> *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

> *Objects*. A package contains a set of related classes.

parameter

> *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

> *Objects*. In an InterSystems IRIS 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

> *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

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 InterSystems IRIS partition or user partition. For UNIX®, see also Raw Disk Partition.

passing by reference

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

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

permission

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

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

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

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

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

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

primary persistent superclass

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

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

principal device

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 InterSystems IRIS. 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

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

priority

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

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

privilege

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

System. A set of one or more InterSystems IRIS 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

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

process-private section

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

process

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

Product Activation Key

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

programmer mode

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

project

> *System*. A user-defined list of related source documents (such as routines or class definitions). Visual Studio Code uses projects to help organize application development.

property

> *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.

property collation

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

property method

> *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.

public

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

1.17 Terms Beginning with Q

query

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

query interface

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

1.18 Terms Beginning with R

range indicator

> *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

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

RECEIVE

> *System*. An InterSystems IRIS system process that broadcasts InterSystems IRIS network configuration information to all known remote computers.

reconfiguration

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

recovery

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

redirection

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

reference

> *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

> *Objects*. A registered class is derived from the %RegisteredObject class. InterSystems IRIS 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

> *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

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

replicated global

>*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

>*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

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

resource

>*System*. The smallest granular unit that can be protected with InterSystems 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

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

role

>*System*. An entity within InterSystems 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

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

roll back

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

roll forward

>*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

>*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

>*System*. In InterSystems IRIS, an ObjectScript program. Routines are stored in an InterSystems IRIS database.

routine buffers

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

row

> *InterSystems 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.

Row ID

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

row specification

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

1.19 Terms Beginning with S

schema evolution

> *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

> *System*. The user with which InterSystems IRIS connects to an LDAP server and whose privileges allow for searching the LDAP database. Its value is specified in either the **LDAP search user DN** or **LDAP username to use for searches** field on the **LDAP Configurations** page (**System Administration** > **Security** > **System Security** > **LDAP Configurations**. (Note that the page name and menu choice includes Kerberos, if that is enabled.)

secondary volume

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

Security Advisor

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

security domain

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

selectivity

> *InterSystems 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

> *System*. See Incoming Lock.

server master

> *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 an InterSystems IRIS process.

server process

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

service

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

shallow save

> *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

> *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

> *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.

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.

special variable

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

SQL

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

SQLCODE

> *InterSystems 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.

SQL computed field

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

SQL mapping

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

SQL query

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

SQL reserved words

InterSystems SQL. SQL reserved words have special meaning in InterSystems 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.

status code

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

storage definition

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

storage interface

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

storage strategy

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

stored procedure

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

stream interface

Objects. The InterSystems IRIS stream interface is used to manipulate streams in ObjectScript, SQL, and Java.

stream

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

superserver

System. The server that listens on a specified port for incoming connections to InterSystems IRIS and dispatches them to the appropriate subsystem. The default superserver port is 1972; if 1972 is not available, the superserver listens on the next available port, starting with 51773. 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

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

system class

> *Objects*. The classes that provide built-in functionality to InterSystems IRIS.

system configuration

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

system manager's directory

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

system name

> *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

> *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

table

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

target role

> *System*. Within a secured InterSystems IRIS 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

> *System*. A user who is attempting to authenticate to an LDAP server. InterSystems IRIS attempts to locate this user in the LDAP database by using the value supplied for the **LDAP Unique search attribute** field on the **Edit LDAP Configuration** page for a particular LDAP configuration. The **Edit LDAP Configuration** page is accessible from the **LDAP Configurations** page (**System Administration** > **Security** > **System Security** > **LDAP Configurations**).

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.

temporary global

System. A global that is stored in the temporary database, IRISTEMP. See Temporary Globals and the IRISTEMP Database.

Terminal

System. Formally, this term refers to the Windows Terminal application. Informally, this term can also refer to the ObjectScript shell, the interactive command line interface to InterSystems IRIS. The Terminal application includes the ObjectScript shell but also provides menus and additional options. See Using the Terminal and Using the ObjectScript Shell.

transaction

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

transient property

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

translation methods

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

trigger

InterSystems SQL. A sequence of actions defined by the developer to execute at various points during an InterSystems SQL application or an object application using InterSystems 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

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 InterSystems IRIS MSU utility to assign a UIC to each database.

unauthenticated access

System. Access to InterSystems IRIS that is not based on any verification of user identity.

Unicode

System. An InterSystems IRIS 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.

unique index

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

unmounted

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

user account

>*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

>*System*. An InterSystems IRIS interactive process that is associated with a specific terminal at which the user has signed on to InterSystems IRIS.

1.22 Terms Beginning with V

valid subscript ranges

>*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 InterSystems IRIS to define an upper boundary on the range.

variable

>*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

>*InterSystems SQL*. A view presents data from one or more tables that meets a specified criteria.

virtual field

>*InterSystems 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

>*InterSystems 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.

1.23 Terms Beginning with W

watchpoint

> *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.

web application

> *System*. A web-based application that can be registered with InterSystems IRIS and thereby regulated as part of the InterSystems security model. The user of a web application must be authenticated to InterSystems IRIS using one of the supported authentication mechanisms.

working set

> *System*. The data typically cached in memory by a given application workload. See database cache.

write image journaling

> *System*. A way to update an InterSystems IRIS 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 production-related classes.