



# InterSystems IRIS Basics: Globals and Multi-Model Data Access

Version 2023.1  
2023-05-25

*InterSystems IRIS Basics: Globals and Multi-Model Data Access*  
InterSystems IRIS Data Platform Version 2023.1 2023-05-25  
Copyright © 2023 InterSystems Corporation  
All rights reserved.

InterSystems®, InterSystems IRIS®, IntegratedML®, InterSystems HealthShare®, InterSystems HealthShare Care Community®, HealthShare Unified Care Record®, InterSystems Caché®, and InterSystems Ensemble® are registered trademarks of InterSystems Corporation. InterSystems IRIS for Health™ and HealthShare® CMS Solution Pack™ are trademarks of InterSystems Corporation.

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

- InterSystems IRIS Basics: Globals and Multi-Model Data Access..... 1**
  - 1 What Are Globals? ..... 1
  - 2 Why Should Application Developers Learn About Globals? ..... 2
  - 3 Accessing Globals with InterSystems IRIS APIs ..... 2
  - 4 Learn More About Globals ..... 2
    - 4.1 Globals and Their Structure ..... 2
    - 4.2 InterSystems Native SDKs ..... 2



# InterSystems IRIS Basics: Globals and Multi-Model Data Access

This article introduces you to the concept of globals, the underlying storage structure for the InterSystems IRIS® data platform. No matter how you decide to store or access your data, what you're doing is using globals.

Globals can be accessed using a relational model, using an object model, or directly. For a video that explains the benefits of this multi-model access and a hands-on exercise that lets you try the three alternatives yourself, see [Exploring Multiple Data Models with Globals](#).

## 1 What Are Globals?

One of the hallmarks of the InterSystems IRIS data platform is its ability to store data once and allow you to access it using multiple paradigms. For example, you can use InterSystems SQL to visualize your data as rows and columns, or you can use ObjectScript and think of your data in terms of objects that have properties and methods. Your application can even mix both data models, using whichever model is easiest and more efficient for a given task. But no matter how you write or access your data, InterSystems IRIS stores it in underlying data structures known as globals.

Globals can be thought of as persistent multidimensional sparse arrays:

- *Persistent* — Globals are stored in the database and can be retrieved at any time, by any process that can access that database.
- *Multidimensional* — The nodes in a global can have any number of subscripts. These subscripts can be integers, decimal numbers, or strings.
- *Sparse* — Node subscripts do not have to be contiguous, meaning that subscripts without a stored value do not use any storage.

Nodes in a global can store many types of data, including:

- Strings
- Numeric data
- Streams of character or binary data
- Collections of data, such as lists or arrays
- References to other storage locations

Even the server-side code you write is ultimately stored in globals!

## 2 Why Should Application Developers Learn About Globals?

While it is possible to write an application on the InterSystems IRIS platform with little or no knowledge of globals, there are several reasons why you may want to learn more about them:

- Some operations may be easier or more efficient if you access globals directly.
- You may want to create custom data structures for data that does not conform to relational or object data models.
- Some system administration tasks are done at the global level, and understanding globals will make these tasks more meaningful to you.

## 3 Accessing Globals with InterSystems IRIS APIs

If you are writing an application in Python, Java, .NET languages, Node.js, or C++, InterSystems IRIS provides APIs that allow you to manipulate your data using the three models discussed in this article, as follows:

- Relational access through [JDBC](#), [ADO.NET](#), DB-API, or ODBC
- Object access through the InterSystems XEP APIs for Java and .Net
- Direct access to globals through the InterSystems Native SDKs

**Note:** Not all forms of access are supported for all languages.

## 4 Learn More About Globals

Use the resources listed below to learn more about globals and how to access them.

### 4.1 Globals and Their Structure

- [Exploring Multiple Data Models with Globals](#) — Video and hands-on exercise
- [Using Globals](#) — Discusses the structure of globals, how to manage them, and how to access them using SQL or ObjectScript.
- [Globals](#) — This section in *Defining and Using Classes* explains the naming conventions for globals created for classes that extend the %Persistent class.

### 4.2 InterSystems Native SDKs

- [Using the Native SDK for Java](#) (interactive course) — Shows you how to use the Native SDK for Java to access globals and call class methods and routines.
- [Using the Native SDK for Java](#) (document) — Shows you how to use the Native SDK for Java to access globals and call class methods and routines.

- [Using the Native SDK for .NET](#) (interactive course) — Shows you how to use the Native SDK for .NET to access globals and call class methods and routines.
- [Using the InterSystems Native SDK for .NET](#) (document) — Shows you how to use the Native SDK for .NET to access globals and call class methods and routines.
- [Using the Native SDK for Node.js](#) — (document) Shows you how to use the Native SDK for Node.js to access globals and call class methods and routines.
- [Using the Native SDK for Python](#) (interactive course) — Shows you how to use the Native SDK for Python to access globals and call class methods and routines.
- [Using the Native SDK for Python](#) (document) — Shows you how to use the Native SDK for Python to access globals and call class methods and routines.

