



Introducing InterSystems Data Fabric Studio in Supply Chain

Version 2.10
2025-03-13

Introducing InterSystems Data Fabric Studio in Supply Chain

PDF generated on 2025-03-13

InterSystems Supply Chain Orchestrator™ Version 2.10

Copyright © 2025 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 Congress Street, Boston, MA 02114, 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

Table of Contents

1 Introduction to InterSystems Data Fabric Studio with supply chain module	1
1.1 Design and Key Features	1
1.2 Users and Where to Start	2
1.3 See Also	2
2 Supply Chain Data Model	3
2.1 Supply Chain Data Objects	3
2.1.1 Master and Reference Supply Chain Data	3
2.1.2 Outbound and Manufacturing Data	3
2.1.3 Inbound Supply Chain Data	4
2.2 Introduction to Data Model Classes	4
2.3 Customizing the Data Model	4
2.3.1 Customizing via Coding	4
2.3.2 Customizing via the API	4
2.4 See Also	5
3 Supply Chain Data Model API	7
3.1 List of Supply Chain Data Objects	7
3.2 Get Object Definition	7
3.3 Adding Custom Attributes	8
3.4 See Also	8
4 Introduction to the Data API	9
4.1 API URL Patterns	9
4.2 Search APIs	10
4.3 Sorting of Results	11
4.4 Default Sorting	11
4.5 Pagination of Results	12
4.6 Date and Date/Time Formats	12
4.7 See Also	13
5 API Reference	15
5.1 Available APIs	15
5.2 See Also	15

1

Introduction to InterSystems Data Fabric Studio with supply chain module

InterSystems Data Fabric Studio™ with supply chain module is a fully managed solution that enables you to centralize data about your supply chain in an easy-to-use low-code environment. You can feed this data to downstream applications, including [InterSystems Supply Chain Orchestrator™](#).

1.1 Design and Key Features

InterSystems Data Fabric Studio™ supports a division of labor in which those who are familiar with the data sources can catalog, label, and describe the available data, so that others can use that data without requiring that deeper knowledge. This division of labor also means that the technical aspects of transforming, validating, and reconciling the data are automated by those who are knowledgeable with those specific requirements, so that others can readily and directly use the resulting data.

The product is also designed to automate all data processing operations, with detailed control over scheduling. Manual and test options are provided for use before any changes are put into production usage.

When combined with the supply chain module, the product provides the following key features:

- [Extensible supply chain data model](#), a canonical supply chain data model implemented on InterSystems IRIS data platform, which can be extended/customized via the data model API.
- [Data model APIs](#) for data model discovery and live documentation, such as listing all supply chain data objects, or getting the detailed definition of a supply chain data object.
- [Data access APIs](#), which support both CRUD operations and advanced search capabilities with sorting and pagination support.
- The ability to define connections to data sources, including safely storing necessary credentials.
- The ability to define schemas or data structures and manage their versions. This includes specifying how data is to be extracted (such as whether only new data is extracted), specifying data types, and specifying default values.
- The ability to define data pipeline recipes that extract data from these external data sources and update tables within Data Fabric Studio. A recipe can define which fields to extract, how to transform the fields if necessary, how to validate the values in the fields, how to reconcile the values with alternative sources of the same data, and finally how to publish the data to a final destination.
- The ability to automate and schedule the running of the recipes, following the appropriate business calendar.

To simplify scheduling, the product supports a hierarchical system of entities, each of which can have its own business calendar but can inherit calendar details from its parent. An entity can correspond to a business unit or can simply correspond to some external system or organization whose calendar is important to your organization.

- The ability to define Business Intelligence cubes based on the tables within Data Fabric Studio. The product provides a built-in analytics tool (InterSystems IRIS® Advanced Analytics), but other Business Intelligence systems can also be used.
- The ability to automate and schedule the building of Business Intelligence cubes.
- The ability to define snapshots of data for review by regulators or analysts. A snapshot can use one or more tables and it generally provides a flattened (de-normalized) view of the relevant data. As with other items, the product provides the ability to automate and schedule snapshots, so that you can accumulate a series of snapshots of the same data. And you can easily define Business Intelligence cubes based on them, for a longitudinal view of that data.

1.2 Users and Where to Start

The product has three general categories of users, each with a different starting place:

- *Administrators*, who perform a small set of administrative tasks, related to security, data sources (at a high level), and system defaults. A key first step is defining an initial set of users and data sources, so that others can start work. See [Welcome, System Administrators](#).
- *Data Engineers* or *Data Stewards*, who define the data pipeline—a generic phrase that refers to defining and cataloging the schemas to be used within the system, defining recipes that load data, and scheduling the recipes. See [Welcome, Data Engineers](#).
- *Data Analysts*, who use the data in the system to build cubes and connect reporting tools. See [Welcome, Data Analysts](#).

1.3 See Also

- [Leveraging InterSystems Data Fabric Studio in Supply Chain](#)
- [Video walkthrough of the InterSystems Data Fabric Studio user interface](#)

2

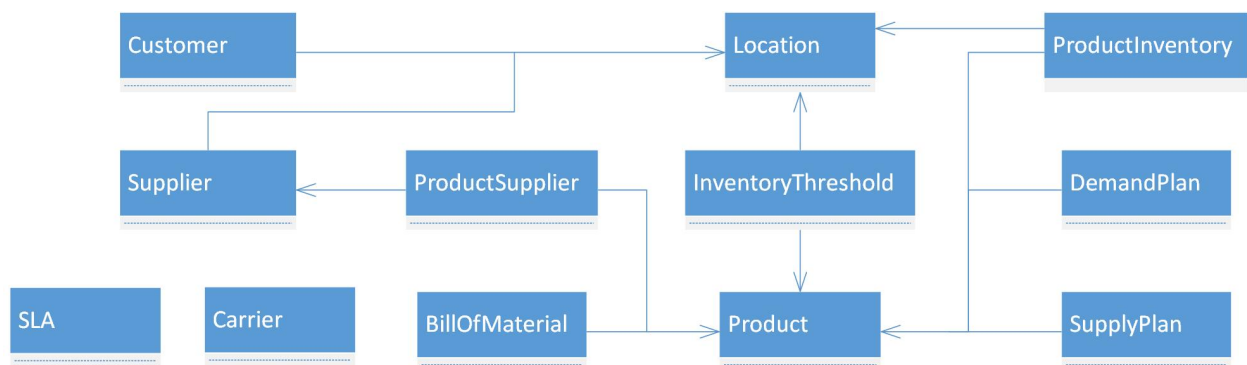
Supply Chain Data Model

This page introduces the supply chain data model.

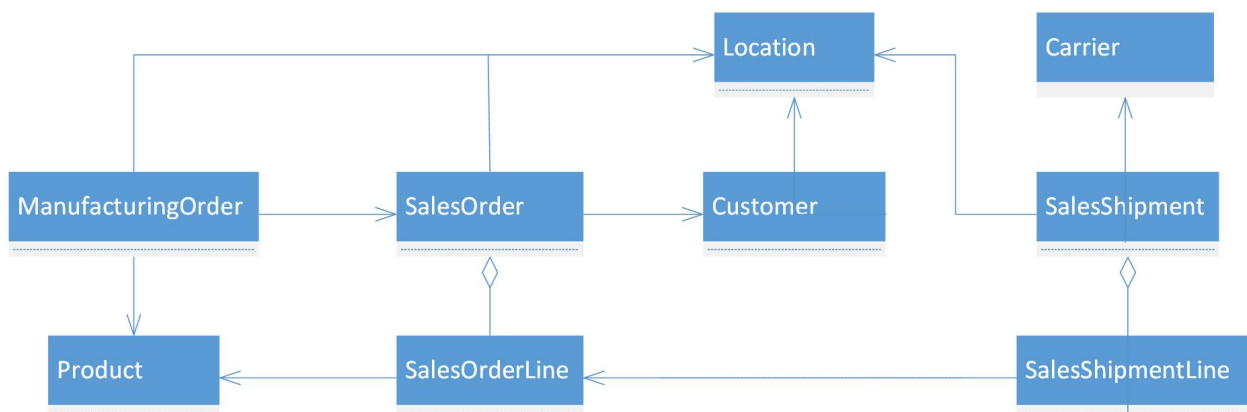
2.1 Supply Chain Data Objects

The following diagrams show the data objects supported in the model, and their relationships.

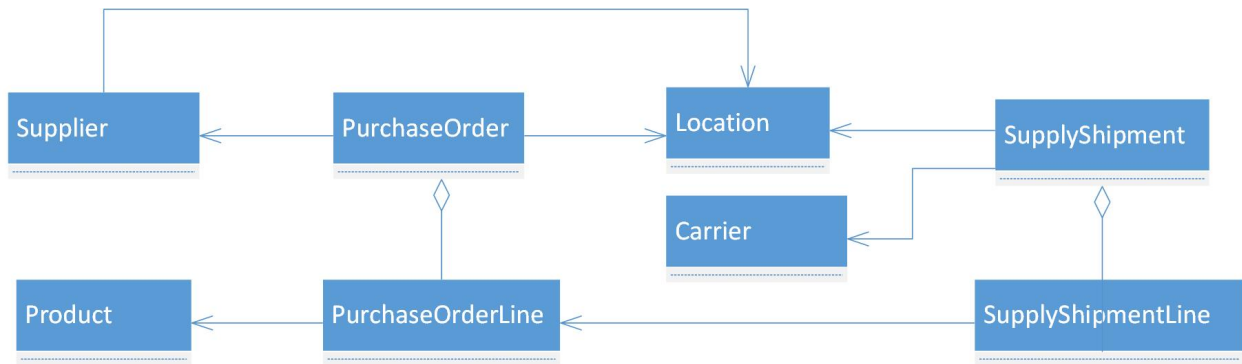
2.1.1 Master and Reference Supply Chain Data



2.1.2 Outbound and Manufacturing Data



2.1.3 Inbound Supply Chain Data



2.2 Introduction to Data Model Classes

All supply chain objects supported by the model are in the package `SC.Data`. These classes can be [customized](#) to suit the business needs of any clients. These data model classes leverage core implementations provided in the package `SC.Core.Data`, but no client should change anything in the core package. To help the consumption of the data model (such as data integration, UI development, and so on), [APIs](#) are provided as live documentation of the data model.

2.3 Customizing the Data Model

You can extend the data model in different ways, depending on your technical skills.

2.3.1 Customizing via Coding

All supply chain data objects have an InterSystems class definition in the package `SC.Data`. These are classes used for all supply chain applications and services. These classes can be modified to add new attributes to the existing data model. As shown in the example below, these classes are mostly empty out of the box because the standard model attributes are all defined in the parent class. You can add new properties or indexes to these class for customization.

```
///This is a custom implementation of the Customer class
Class SC.Data.Customer Extends SC.Core.Data.Customer
{
    Property MyNewProp as %String;
}
```

2.3.2 Customizing via the API

For people without class programming knowledge, or who just need to make a change without direct access to the IRIS server, there is an [API call](#) you can use to add new custom attributes to any supply chain object. As a result of such API calls, the corresponding object definition classes are updated (as if manually coded), database tables are adjusted automatically, and the attributes are available to be used anywhere on the InterSystems IRIS® platform, such as in analytics cubes, or in a business process.

2.4 See Also

- [Supply Chain Data Model API](#)
- [Supply Chain Data API](#)
- [API Reference](#)

3

Supply Chain Data Model API

This page introduces the supply chain data model API. You can use this API for model discovery and customization, including ways to

- List all supply chain data objects
- Retrieve details of any objects, with attributes, data type, size limit, and so on
- Extend the data model by adding custom attributes to existing supply chain objects

3.1 List of Supply Chain Data Objects

Use this API call to find all supply chain objects in the data model:

```
GET {{IRIS-SERVER}}/api/scdata/v1/objects
```

The response looks like the following example:

```
[
  {
    "objectName": "BOM",
    "className": "SC.Data.BOM",
    "description": "Object for bill of material. This object is commonly
                  used in manufacturing to specify the parts required
                  to make or assemble a product."
  },
  {
    "objectName": "Carrier",
    "className": "SC.Data.Carrier",
    "description": "This object is used to capture shipment
                  carrier information."
  },
  ...
]
```

To find the details of each object, use the next API call with the `objectName` value returned in this API response.

3.2 Get Object Definition

To get the details of an object, first get the object name using the above API call. Then use this API call:

```
GET {{IRIS-SERVER}}/api/scdata/v1/objects/[ObjectName]
```

For example, the following is part of the response returned for Customer object:

```
{
  "objectName": "Customer",
  "className": "SC.Data.Customer",
  "objectName": "This object is used to capture the master
    data for a customer.",
  "attributes": [
    {
      "name" : "uid",
      "description" : "unique ID of a customer",
      "dataType" : "String",
      "required" : 1,
      "maxLength" : "256",
      "isCustom" : 0
    },
    {
      "name" : "name",
      "description" : "official name of the customer",
      "dataType" : "String",
      "required" : 0,
      "maxLength" : "256",
      "isCustom" : 0
    },
    ... ]
}
```

3.3 Adding Custom Attributes

To add a new attribute to an object in the supply chain model, use this API call:

```
POST {{IRIS-SERVER}}/api/scdata/v1/attributes/[ObjectName]
```

with the attribute definition JSON in the API body, such as:

```
{
  "name" : "customProperty",
  "description" : "Custom attribute added for testing",
  "dataType" : "String",
  "maxlength" : 120,
  "required" : 1
}
```

Once an attribute is added through this API, data access APIs can be used to load data into the attributes, or search for records based on the newly created attributes. No additional step or waiting is needed.

Important: This API does not support more advanced customizations as can be made by modifying the class or table definitions directly, such as adding indexes or validation rules. Also, there is no API to update or delete custom attributes, so use this API with caution.

3.4 See Also

- [Supply Chain Data Model](#)
- [Supply Chain Data API](#)
- [API Reference](#)

4

Introduction to the Data API

The supply chain data [API](#) enables you to create, update, delete, and retrieve data. This page provides an introduction to this API.

4.1 API URL Patterns

All the API calls follow the same URL pattern:

```
GET {{IRIS-SERVER}}/{{DATAMODEL-PATH}}/OBJECT_PATH?parameters
POST {{IRIS-SERVER}}/{{DATAMODEL-PATH}}/OBJECT_PATH
GET/PUT/DELETE {{IRIS-SERVER}}/{{DATAMODEL-PATH}}/OBJECT_PATH/uid
```

Where:

- The `{{IRIS-SERVER}}` part is the server information of the underlying InterSystems IRIS® instance.
- The `{{DATAMODEL-PATH}}` part is the API base URL, such as `/api/scdata/v1`
- The `OBJECT_PATH` part is simply the object name in lower case plural form, such as `salesorders` or `customers`. See the [table below](#) for all object values.
- The first URL pattern gets data for an object, or searches for objects using a set of parameters. Any attributes of the object can be used in a [search criterion](#).
- The second URL pattern creates a new object record. The body of the request should include the JSON string for the new object.
- The third URL pattern retrieves, updates, or deletes an object record by its `uid` (which is the external primary key).

Some examples:

- Finding all sales orders with status `Open` or `PartialShip`:

```
GET {{IRIS-SERVER}}/{{DATAMODEL-PATH}}/salesorders?orderStatus=Open,PartialShip
```

- Creating a new customer:

```
POST {{IRIS-SERVER}}/{{DATAMODEL-PATH}}/customers
```

with the following JSON in the request body:

```
{
  "uid" : "CUST-TEST-101",
  "name" : "Google",
  "type" : "HighTech",
  "contact" : "Ming",
  "url" : "https://google.com"
}
```

- Retrieving a supply shipment record with uid value SUP-SHIP-1001:

```
GET {{IRIS-SERVER}}/{{DATAMODEL-PATH}}/supplyshipments/SUP-SHIP-1001
```

- Updating a location record:

```
PUT {{IRIS-SERVER}}/{{DATAMODEL-PATH}}/locations/LOC-PLANT-002
```

with new location JSON data in the request body.

4.2 Search APIs

Searches are supported through GET APIs with one or multiple URL parameters; each parameter maps to one search condition. If more than one parameter are used, the parameter conditions are combined with logical AND operation. For example, the following searches all Lenovo laptop products:

```
GET /products?brand=Lenovo&category=laptop
```

Each parameter name should match exactly an attribute name of the primary object. In some cases, attributes from secondary objects (contained or referenced objects) can also be used. For example, some product attributes can be used in an order search.

The search parameter value can be a single value, a list of values, or a range of values, as follows:

- *Single value.* This is the simplest search criterion, using the format `parameterName=value`. If the parameter is a string or a list of strings, the value matching is case-insensitive. So `brand=lenovo` is the same as `brand=Lenovo`.
- *List of values.* For a condition that requires matching any of the values in a list, you can provide a list of values as a comma-separated string. For example, to find products of either Lenovo brand or Dell brand, you can use `brand=Lenovo,Dell`. The list can be used for all data types, string, date, or numbers.
- *Range of values.* For date and numerical values (including currency), a range can be used in a search parameter. A range is specified by two values, separated by `..` (two dots), in format of `min..max`. Both values are inclusive for the range. For example, `price=100..200` finds all records with price in range of 100 (including 100) and 200 (including 200). The two range boundary values are not always required, and if one is missing, it means that boundary does not exist. For example, `price=..200` means any record with price less than or equal to 200, while `price=100..` means any record with price greater than or equal to 100.
- *Null value.* For any data type, you can search for records with an attribute not set (that is, a null value), or an attribute with some value set (that is, not null). For such cases, you can use two special string values `NULL` and `NOTNULL` for any attribute, such as `attr1=NULL&attr2=NOTNULL`

For example, this call finds all laptop computers from either Lenovo or Dell with price range between 500 and 1000:

```
GET /products?category=laptop&brand=Lenovo,Dell&price=500..1000
```

4.3 Sorting of Results

When more than one record is returned by an API call, you can define the order of the response data by specifying the sorting parameter `sortBy`. The value of this parameter must be a comma-separated list of attribute names of the primary object returned.

For example, the following API call defines how a list of orders should be sorted in the response:

```
GET /salesorders?sortBy=customer,orderValue
```

which defines the sorting to be first by customer, and then by order value.

By default, sorting is done in ascending order. To change the order to descending, use the `-` sign before the attribute name, such as the following example, which first sorts by customer in ascending order, and then by `orderValue` in descending order:

```
GET /salesorders?sortBy=customer,-orderValue
```

4.4 Default Sorting

If no sorting parameter is provided in the request URL, a default sorting is applied, based on the primary object returned. The following table lists the default sorting parameters for each object type:

Object	URL path	Default sorting attributes
Carrier	carriers	name
Customer	customers	name
Supplier	suppliers	name
Product	products	name
Location	locations	locationName
BillOfMaterial	billofmaterials	productId, parentItemId
InventoryThreshold	inventorythresholds	siteLocationId, productId
Milestone	milestones	ID
ProductInventory	productinventories	siteLocationId, productId
ProductSupplier	productsuppliers	productId
SupplyPlan	supplyplans	locationId, productId
DemandPlan	demandplans	locationId, productId
SalesOrder	salesorders	-orderPlacedDate
SalesOrderLine	salesorderlines	orderId, lineNumber
SalesShipment	salesshipments	-actualShipDate
SalesShipmentLine	salesshipmentlines	shipmentId, lineNumber
PurchaseOrder	purchaseorders	-orderPlacedDate

Object	URL path	Default sorting attributes
PurchaseOrderLine	purchaseorderlines	orderId, lineNumber
SupplyShipment	supplyshipments	-actualShipDate
SupplyShipmentLine	supplyshipmentlines	shipmentId, lineNumber
ManufacturingOrder	manufacturingorders	-orderEntryDate

4.5 Pagination of Results

By default, any API call that returns multiple records returns the first 100 records that match the given criteria. You use pagination parameters to obtain additional records or a larger set of records. The pagination parameters give you the control needed to, for example, populate a table in a web UI, without excess resource consumption (memory, network bandwidth, and so on) or performance issues.

The pagination parameters are as follows:

- **pageSize.** This parameter defines the maximum number of record returned in the API call. The actual number of records returned can be less or equal to this value. The default value is 100, and the maximum allowed value is 1000.
- **pageIndex.** This parameter specified the page index (starting from 0), to be returned in the response. The default is 0.

For example, the following API call skips the first 200 orders and returns the next 100 orders, sorted by order value:

```
GET /salesorders?sortBy=orderValue&pageSize=100&pageIndex=2
```

When a response returns only partial records using pagination, the following HTTP header parameters are populated in the response. Use these values to control the pagination logic in the UI.

- **pageSize.** The same value if specified in the request path parameter. Returns the default value if not explicitly specified in the request.
- **pageIndex.** The same value if specified in the request path parameter. Returns the default value if not explicitly specified in the request.
- **returnCount.** Number of records returned in the current response. This value is less than or equal to **pageSize**

4.6 Date and Date/Time Formats

For any date or date/time attribute, you must specify the value in ISO 8601 format, whether you provide this in the message JSON body or in the HTTP parameter.

Here are some examples. For date attributes:

```
2021-02-28
```

For date/time attributes (the following two values are equivalent):

```
2021-12-15T13:23:15-05:00
2021-12-15T18:23:15Z
```


If the time zone information is missing, the UTC time is assumed. For example, 2021-05-24T08:30:00 is treated the same as 2021-05-24T08:30:00Z

4.7 See Also

- [Supply Chain Data Model](#)
- [API Reference](#)

5

API Reference

This page provides links to references for APIs that are relevant when you are creating a solution with InterSystems Data Fabric Studio™ with supply chain module.

5.1 Available APIs

InterSystems Data Fabric Studio in supply chain includes two APIs:

- [Data Model API](#) — Use this to browse the supply chain data model; extend the data model.
- [Data API](#) — Use this to create, update, and delete supply chain data; create issues.

5.2 See Also

- [Introduction to the Data Model API](#)
- [Introduction to the Data API](#)

