InterSystems™
IRIS Data Platform

Using the Simple Data Transfer Utility

Version 2020.3
2021-02-04

InterSystems Corporation   1 Memorial Drive   Cambridge MA 02142   www.intersystems.com

Using the Simple Data Transfer Utility
InterSystems IRIS Data Platform Version 2020.3 2021-02-04

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

Using the Simple Data Transfer Utility

The Simple Data Transfer Utility is a Java command-line utility used for massive data transfer from a JDBC data source or CSV file to a JDBC compliant database. The utility is implemented in a single Java class. While you can use any supported target or source, the utility is optimized to work with InterSystems IRIS as the target.

The utility is intended primarily for extremely fast relocation of huge datasets. While it may function successfully even if actions such as INSERT or DELETE are done in parallel, it is not designed to be an integral part of a typical production environment. In an optimal setup, no other database processes are running, journaling is turned off, and some concurrency controls may be disabled while the transfer is taking place.

Furthermore, the utility works with both standard and sharded namespaces in InterSystems IRIS, and takes full advantage of parallelization when the target table is sharded.

This article discusses the following topics:

1 Preparing Your Environment

The Simple Data Transfer Utility requires Java 8 and the following JAR files, which InterSystems provides in the InterSystems IRIS root directory:

- intersystems-jdbc-3.0.0.jar — Contains the core classes required for JDBC connections to InterSystems IRIS.
- intersystems-utils-3.0.0.jar — Contains the com.intersystems.datatransfer.SimpleMover class, which implements the utility.
- intersystems-xep-3.0.0.jar — Contains the classes required to transfer data using XEP functionality. For an overview of XEP, see First Look: Java Object Persistence with XEP.

Additionally, if the source or target of the data transfer is a JDBC data source that is not provided by InterSystems, the utility requires the JAR file for the corresponding JDBC driver.

For more information about InterSystems JAR files, see The InterSystems Java Class Packages.

2 Launching the Utility

You launch the Simple Data Transfer Utility from the command line as follows.

1. Open a command prompt.

2. Navigate to the location of the properties file that you created for the utility.

   For instructions on setting up the properties file, see Configuring the Utility.

3. Issue the command to launch the utility.

   If the source and target of the data transfer are InterSystems databases, or if the source is a CSV file and you intend to use the built-in JDBC driver for CSV files, you issue the following command:

   ```
   java -cp <jdbc_path>/intersystems-utils-3.0.0.jar; \
            <jdbc_path>/intersystems-jdbc-3.0.0.jar; \
            <jdbc_path>/intersystems-xep-3.0.0.jar \
            com.intersystems.datatransfer.SimpleMover p=example.properties
   ```

   where `<jdbc_path>` is the location of the InterSystems JAR files on your system, `;` is the classpath separator that is appropriate for your system, and `example.properties` is the name of the properties file.

   If the source or target of the data transfer is a JDBC data source or JDBC-compliant database that is not provided by InterSystems, or if the data source is a CSV file and you intend to use a third-party JDBC driver for CSV files, you must also specify the JAR file for the corresponding JDBC driver.

   Note: Line breaks have been added for clarity and should not be used in practice.

   The utility writes progress to a log file.

3 Configuring the Utility

You configure the Simple Data Transfer Utility using a properties file, which you reference when you launch the utility.

At minimum, the file must identify the source and target of the data transfer. However, the file can also specify other details of the transfer such as the number of records to load at one time. For example, the following properties file defines the example01.csv file as the source of a data transfer, the ExampleOne table on a local InterSystems IRIS instance as the target, and the example01log.txt file as the location for logging information:

```
source.url=file:/some/path/example01.csv
source.type=csv

target.url = jdbc:IRIS://localhost:56777/USER
target.username=_SYSTEM
target.password=SYS

target.table=ExampleOne

log.file=file:/some/path/example01log.txt
```

For additional examples, see Example Properties Files. To review the options for logging the utility's progress, see Logging and Monitoring Properties.

You can optionally override one or more of the properties in the file using command-line arguments. However, InterSystems recommends that you do so only for testing purposes.

Broadly, the properties fall into the following categories:

- Required Properties
- Source Properties
- Target Properties
- Job Control Properties
- Logging and Monitoring Properties

3.1 Required Properties

Only the following five properties may be required:

- source.url
- source.table or source.query, if the source is a JDBC data source
- source.header, if the source is a CSV file
- target.url

3.2 Source Properties

Source properties identify the data to transfer.

The following table lists the source properties that apply to both JDBC data sources and CSV files.

Property	Required	Description

source.url	Yes	URL for the JDBC data source or file path to the CSV file, for example:
		`IRIS://localhost:1972/USER`
		or
		`file:/tmp/patients.csv`
		For InterSystems IRIS sources, you can specify the URL differently. For more information, see Specifying an InterSystems Source or Target.

source.type	No	Type of JDBC driver to use: • `csv`—Driver for CSV files built in to InterSystems IRIS • `jdbc`—Any other JDBC driver The default value is `jdbc`.

3.2.1 JDBC Data Source Properties

The following table lists the source properties that apply only to JDBC data sources.

Property	Required	Description
source.username	No	Username credential required to gain access to the data source
source.password	No	Password credential that corresponds to the source.username value
source.table	Yes, if you do not set source.query	Table to transfer data from
source.query	Yes, if you do not set source.table	SQL query used to load data from the source, for example: `source.query = select * from $table where filteredOut = 0 and ID between ? and ?` For more information, see Specifying an SQL Query for Loading Data.

Property	Required	Description
source.query.$1	No	First query parameter, which you can use to split the data into batches For more information, see Specifying an SQL Query for Loading Data.
source.query.$2	No	Second query parameter, which you can use to split the data into batches For more information, see Specifying an SQL Query for Loading Data.
splitOn	No	Name of the column to use to split the source data into chunks For example, you might choose to split the source data into chunks using an ID column.
min	No	Minimum value of the column that you specified in the splitOn property
max	No	Maximum value of the column that you specified in the splitOn property
exclude	No	List of columns from the source table to exclude from the data transfer The utility does not read data from the columns that you specify and does not write data from the columns that you specify to the target table.
source.count	No	Number of records to load at one time when you specify a source.query value The utility uses the source.count value to display the percentage of the data that it has loaded at a given time. You can set either the source.count orsource.query.count property.
source.query.count	No	SQL query used to derive the number of records to load when you specify a source.query value The utility uses the source.query.count value to display the percentage of the data that it has loaded at a given time. You can set either the source.query.count or source.count property. The default value is `SELECT COUNT(*) FROM (source.query)`.
source.execute.before	No	SQL statements or (for InterSystems IRIS data sources only) ObjectScript class methods to execute on the source data prior to the data transfer
source.execute.after	No	SQL statements or (for InterSystems IRIS data sources only) ObjectScript class methods to execute on the source data after successful completion of the data transfer
source.driver.property	No	Property and value pair that is passed directly to the JDBC driver For example, if you specify **`source.driver.propertya=25`**, then the utility passes a property with the name propertya and a value of `25` to the JDBC driver for the source.

3.2.2 CSV Source Properties

The following table lists the source properties that apply only to CSV sources.

Property	Required	Description
source.header	Yes	List of column names to sequentially map to the fields in the CSV file For example, if you specify **`Name`**, **`DOB`**, **`SSN`**, then the utility maps the first field in the file to the Name column, the second field in the file to the DOB column, and the third field in the file to the SSN field. A value of # indicates that the first line in the file is the header.
source.separator	No	Separator used to delimit fields in the CSV file The default value is the comma character (,).
source.jobs	No	Number of parallel jobs to use for reading in the CSV file
source.csv.pool	No	Pool size to grant to the utility for reading in the CSV file Since reading operations are typically faster than write operations, the utility may read an entire CSV file into memory. If the file is very large, doing so may cause an `OutOfMemory` error. Setting a pool size prevents the error. InterSystems recommends that you provide as large a pool as your memory allows. The source.csv.pool value cannot be less than the batch size value multiplied by the thread value. For more information about the latter properties, see Job Control Properties. If you specify a pool size that is smaller than the minimum value, the utility increases the pool size to the minimum value. If you do not specify a pool size, the pool size is unbounded.

3.3 Target Properties

Target properties identify the JDBC compliant database that is the target of the data transfer.

The following table lists the target properties.

Property	Required	Description

Property	Required	Description
target.url	Yes	URL for the target JDBC compliant database: `IRIS://localhost:1972/USER` For InterSystems IRIS targets, you can specify the URL differently. For more information, see Specifying an InterSystems Source or Target.
target.username	No	Username credential required to gain access to the target database
target.password	No	Password credential that corresponds to the target.username value
target.create	No	If the source of the data transfer if a JDBC data source, indicates whether to create a table in the target database that is identical to the table in the source: • `sharded`—Create a sharded version of the table • `not sharded`—Create a version of the table that is not sharded • `do not create`—Do not create the table If the value is `do not create`, the table must exist in the target database. The default value is `do not create`.
target.mode	No	Connection type to use when inserting records into the target database: • `jdbc`—Use the standard JDBC connection • `xep`—Use the InterSystems XEP API. An XEP connection can provide significant performance improvements over a JDBC connection. However, you cannot use an XEP connection if the target.create value is `sharded`. The default value is `jdbc`.
target.monitor	No	Indicates whether a background thread executes a `SELECT COUNT (*)` query on the target table to determine the rate of insertion If you set the value to true, the background thread executes the query. If you set the value to false, the utility estimates the rate of insertion based on the amount of data sent to the server. The default value is `true`.

target.execute.before	No	SQL statements or (for InterSystems IRIS data sources only) ObjectScript class methods to execute prior to transferring data to the target
		For example, you might specify an SQL query to drop a table or delete records from it:
		`target.execute.before=drop table myschema.mytable`
		`##class(myclass).%KillExtent() delete from myclass`

Property	Required	Description
target.execute.after	No	SQL statements or (for InterSystems IRIS data sources only) ObjectScript class methods to execute after successful completion of the data transfer, for example: `target.execute.after=##class(p155618.TT2).%BuildIndices()`
target.driver.property	No	Property and value pair that is passed directly to the JDBC driver For example, if you specify **`target.driver.propertya=25`**, then the utility passes a property with the name propertya and a value of 25 to the JDBC driver for the target.

3.4 Job Control Properties

Job control properties enable you to distribute resources for the data transfer.

The following table lists the job control properties.

Property	Required	Description
batch size	No	Number of records to insert into the target database at one time For example, a value of **`100000`** includes 100000 records in each insert operation.
limit	No	Maximum number of records to transfer For example, if you set the value to **`200000`** and the source contains 300000 records, the utility transfers only 200000 records.
jobs	No	Number of chunks (or jobs) to split the data transfer into For example, a value of **`10`** indicates that the utility should move the source data to the target database in 10 separate chunks
threads	No	Number of chunks of data to transfer at one time The default value is the value of the jobs property.

3.5 Logging and Monitoring

Logging and monitoring properties enable you to specify where and how the utility logs the progress of the transfer.

The following table lists the logging and monitoring properties.

Property	Required	Description
log.file	No	Location for logging information The utility writes metadata about the source and target as well as progress indicators to the file. The default value is the standard output (stdout) location for your system.
refresh	No	Interval in seconds (s) or milliseconds (ms) at which the utility writes the progress of the data transfer to the log file For example, if you set the value to **10ms**, the utility writes progress updates every 10 milliseconds. The default value is 3s.

4 Specifying an InterSystems Source or Target

When the source of the data transfer is an InterSystems database, you can use the following three properties instead of the **source.url** property:

- **source.host**—TCP or IP host URL
- **source.port**—Superserver port number
- **source.namespace**—Namespace that contains the source table

Similarly, when the target of the data transfer is an InterSystems database, you can use the following three properties instead of the **target.url** property:

- **target.host**—TCP or IP host URL
- **target.port**—Superserver port number
- **target.namespace**—Namespace that contains the target table

The utility uses each set of three properties to construct an InterSystems connection URL. For example,

```
source.host=localhost
source.port=1972
source.namespace=SHMASTER
```

is exactly equivalent to

```
source.url = jdbc:iris://localhost:1972/SHMASTER
```

For more information about the InterSystems JDBC connection string, see Defining a JDBC Connection URL.

5 Specifying an SQL Query for Loading Data

When the source of the data transfer is a JDBC data source, the utility runs an SQL query to extract data from the source. You specify the query by setting *one* of the following properties:

- **source.table**—Name of the table to extract data from. The utility runs a `SELECT * FROM` *`table`* query, where *`table`* is the name you specified.
- **source.query**—Custom SQL query.

Note: If you set both the **source.table** and **source.query**, the utility throws an error.

If you use a custom query, you can optionally include a range condition in the query to split the data into batches, for example:

```
source.query = select * from $table where filteredOut = 0 and ID between ? and ?
```

where the question marks (`?`) serve as placeholders for query parameters. You specify the query parameters using the following properties:

- **source.query.$1**—First query parameter, which determines the first value in the range condition.

  You define **source.query.$1** using the ObjectScript format for FOR loop. For example,

  ```
  source.query.$1 = 1:10:100
  ```

  sets the first query parameter for the first batch to 1, and increments the value by 10 (to 21, 31, and so on) for subsequent batches until it reaches 100.

  For more information about formatting the property, see FOR With an Argument.
- **source.query.$2**—Second query parameter, which determines the second value in the range condition.

  You can reference **source.query.$1** when you define **source.query.$2**. For example,

  ```
  source.query.$2 = $1 + 999999
  ```

  sets the second query parameter to the value of **source.query.$1** incremented by 999999 for each batch.

See the next section for an example.

6 Example Properties Files

The following sections provide examples of properties files ingested by the Simple Data Transfer Utility. For information about the properties shown in the examples, see Configuring the Utility..

6.1 PostgresSQL Data Source

The following example property file transfers data from a PostgresSQL table into a table in an InterSystems IRIS database. Note that the source.query property is configured to load the source data in batches.

```
exclude=ID
refresh=5s
log.file=logs/test.log
source.host=localhost
source.port=56775
source.namespace=DEMO
source.username=USER
source.password=PASSWORD
source.table=demo_demo.demo
source.query = select * from $table where filteredOut = 0 and clade->NodeType = 'duplication' and
   ID between ? and ?
source.query.$1 = 1000000:1000000:10000000
source.query.$2 = $1 + 999999
source.query.count = select count(*) from $table where filteredOut = 0 and clade->NodeType =
   'duplication' and ID between 1000000 and 10999999
```

```
target.host=localhost
target.port=1972
target.namespace=TEST
target.username=USER
target.password=PASSWORD
target.table=test_test.test
target.create = not sharded
```

6.2 CSV Data Source in XEP Mode

The following example property file transfers data from a CSV file into InterSystems IRIS using XEP functionality.

```
exclude=dummy
refresh=5s
log.file=logs/test.log
jobs = 4
source.type=csv
source.url=file:/filepath/sample.csv
source.driver.headerline=id,ra,dec,errra,errdec,pmra,pmdec,errpmra,errpmdec,radvel,errradvel,htm,
   healpixring,healpixnest,epoch,axe_a,axe_b,theta,shape,magu,errmagu,magb,errmagb,magv,errmagv,magr,
   errmagr,magi,errmagi,magj,errmagj,magh,errmagh,magk,errmagk,magSg,errmagSg,magSr,errmagSr,magSi,
   errmagSi,vartype,period,logteff,errlogteff,logg,errlogg,logmet,errlogmet,alphamet,erralphamet,
   spectrumid,dummy
source.jobs = 1
source.csv.pool = 200000
target.host=localhost
target.port=1972
target.mode=xep
target.namespace=DEMO
target.username=USER
target.password=PASSWORD
target.table=test
```