InterSystems™
IRIS Data Platform

First Look: Data Transformations

Version 2018.1
2018-11-30

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

First Look: Data Transformations
InterSystems IRIS Data Platform Version 2018.1 2018-11-30

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

Table of Contents

First Look: Data Transformations

This First Look guide helps you transform data with InterSystems IRIS Data Platform™ integration productions.

1 Data Transformation and Interoperability

Data transformation is at the heart of the interoperability of the InterSystems IRIS Data Platform™. You have the power to change the format and content of incoming data from one system to meet the requirements of a downstream system, allowing the systems to communicate with each other. Simply put, messages sent from one system can be transformed into messages that the other application can understand. With InterSystems IRIS, data transformations are simple to create, test, and maintain.

Suppose you have two retail systems that include the price of products. When System A sends data to other systems, it includes the base price of a product without tax added. However, System B needs the price to include taxes for the region. A data transformation in an InterSystems IRIS interoperability production can covert the price received from System A into the tax-added price before the data is sent to System B. Once defined, the data transformation takes care of the modifications automatically.

Common transformations include:

- Copying values from the source message to the target message.
- Performing calculations based on values of the source message and copying the results to the target message.
- Assigning new literal values to the target message.
- Rearranging the order of data.

2 Exploring the DTL Editor

You can write DTL (data transformation language) code to create a data transformation or you can create one using the DTL Editor. The DTL Editor allows non-technical users to create data transformations without having to write code. For example, its graphical user interface allows you to quickly map values from the source message to the target message with a drag-and-drop action. The following tour of the DTL Editor guides you through its major features by creating a data transformation within a production. For an introduction to InterSystems IRIS productions, see First Look: Connecting Systems Using Interoperability Productions.

In this demo, data about a student’s final grade in a class must be converted so that a different application can use the data.

2.1 Before You Begin

Before starting this tour of the DTL Editor, you need to:

- Download an InterSystems IRIS production and sample file from GitHub.
- Install and run InterSystems IRIS.
- Create an interoperability-enabled namespace.

- Import the downloaded production into the namespace.

2.1.1 Downloading from GitHub

You do not need any experience with GitHub to download the production and data file that are used with this demo.

1. Go to https://github.com/intersystems/FirstLook-DataTransformations in a web browser.
2. Click **Clone or download**.
3. Click **Download ZIP**.
4. Extract the files from the zip archive that was downloaded to your machine. The XML file is a representation of an InterSystems IRIS production that you will deploy on your system.

2.1.2 Installing InterSystems IRIS

You need a running, licensed instance of InterSystems IRIS to complete the remaining steps. For instructions on how to install and license a development instance of InterSystems IRIS, see Quick Start: InterSystems IRIS Installation.

2.1.3 Creating an Interoperability-Enabled Namespace

You must create an interoperability-enabled namespace before you can import the production that you downloaded from GitHub. Within InterSystems IRIS, a namespace is a logical entity that provides access to data and code. The namespaces created when you first install InterSystems IRIS are not interoperability-enabled. In the Management Portal:

1. Select **System Administration** > **Configuration** > **System Configuration** > **Namespaces** to go to the **Namespaces** page.
2. On the **Namespaces** page, select **Create New Namespace**. This displays the **New Namespace** page.
3. On the **New Namespace** page, enter the name for the new namespace, such as `DTLdemo`.
4. Next to the **Select an existing database for Globals** drop-down menu, select **Create New Database**. This displays the **Database Wizard**.
5. On the first page of the **Database Wizard**, in the **Enter the name of your database** field, enter the name of the database you are creating, such as `DTLdemodb`. Enter a directory for the database, such as `C:\InterSystems\IRIS\mgr\DTLdemodb`. On that page, select **Next**.
6. On the next page, select **Finish**.
7. Back on the **New Namespace** page, in the **Select an existing database for Routines** drop-down menu, select the database you just created.
8. Ensure that the check box to specify that the namespace is enabled for interoperability productions is selected.
9. Select **Save** near the top of the page and then select **Close** at the end of the resulting log.

2.1.4 Importing Demo Production

Once you have an interoperability-enabled namespace, you can import the demo production that you downloaded from GitHub. This production will contain your data transformation. To import the production:

1. Open the Management Portal from the InterSystems IRIS launcher.
2. Select **Interoperability > Manage > Deployment Changes > Deploy** to go to the **Deploy Production Changes** page. If prompted, select the namespace that you created before completing this step.
3. Click **Open Deployment**.
4. Go to the directory where you downloaded the files from GitHub, select DTLStudentDemo.xml, and click **OK**.

5. Click **Deploy**.
6. Click **OK**.

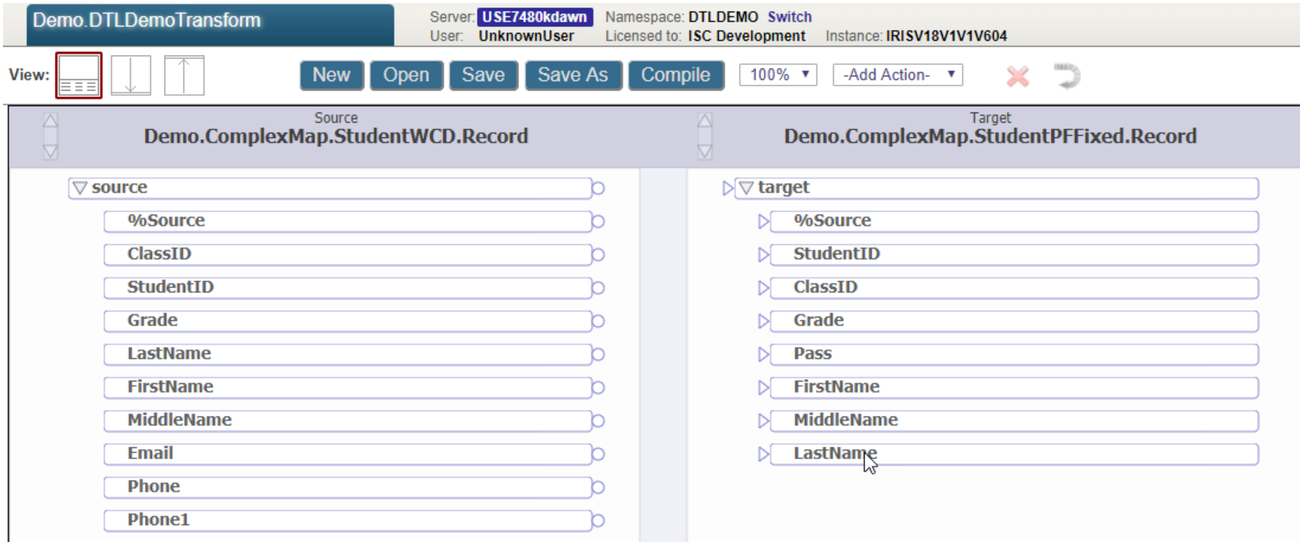
2.2 Creating the Data Transformation

InterSystems IRIS provides a Data Transformation Wizard to speed up and simplify the process of creating a new data transformation. For this demo, you are creating a data transformation that connects two record maps related to a student's grade in a class. To create the data transformation:

1. From the Management Portal home page, select **Interoperability > List > Data Transformations**.
2. Click **New**.

   The Data Transformation Wizard opens.
3. Select **Demo** from the **Package** drop-down list.
4. In the **Name** field, enter a unique name such as `DTLDemoTransform`.
5. Click the magnifying glass next to the **Source Class** field.
6. Click **Message Classes > Demo > Complex Map > StudentWCD > Record**.
7. Click the magnifying glass next to the **Target Class** field.
8. Click **Message Classes > Demo > Complex Map > StudentPFFixed > Record**.
9. Click **OK**.

   The new data transformation opens in the DTL Editor.


Demo.DTLDemoTransform

Server: USE7480kdawn Namespace: DTLDEMO Switch
User: UnknownUser Licensed to: ISC Development Instance: IRISV18V1V1V604

Source: Demo.ComplexMap.StudentWCD.Record

Target: Demo.ComplexMap.StudentPFFixed.Record

2.3 Transforming the Data

Now that you have created the data transformation, use the DTL Editor to:

- Copy data from the source to the target, rearranging the order of the data as needed.
- Add a college-specific code to the beginning of the course number.
- Round off the final grade to a whole number.
- Use the final grade to determine whether the student passed or failed, and indicate this result in the target file.

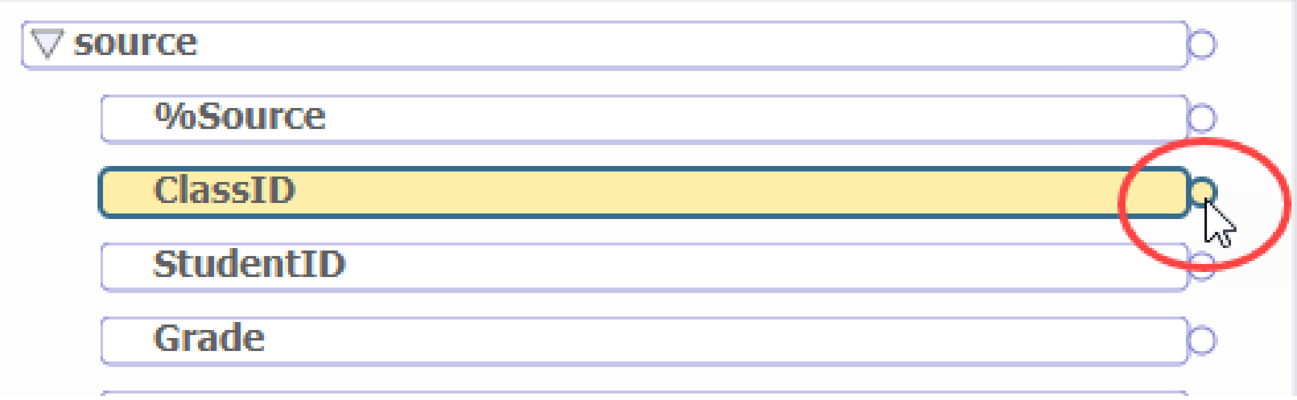
- Ignore data from the source that is not required in the target.

2.3.1 Copying Data

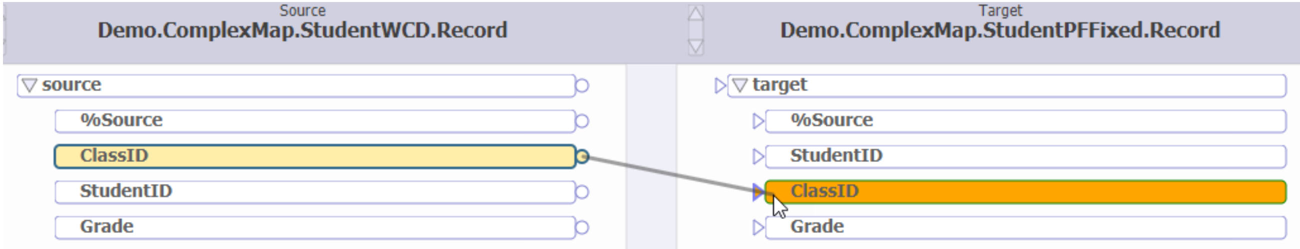
One of the most powerful and time-saving features of the DTL Editor is the ability to drag and drop values from the source to a corresponding property of the target. You simply click and hold the circle on the source's property and drag it to a property of the target message.

To copy values from the source to the target:

1. Click and hold the circle of the ClassID property of the source (on the left-hand side of the Editor).



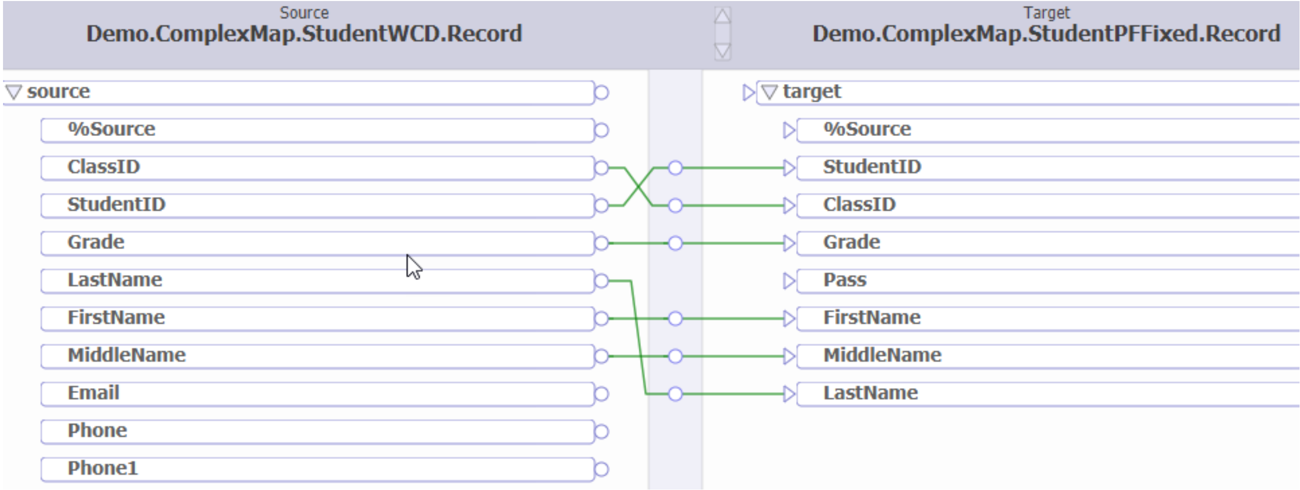
2. Drag your cursor to the ClassID property of the target until it is highlighted, then release the mouse button.



The value of the source's ClassID will be copied to the target's ClassID property.

3. Following this procedure, connect the following source properties to the corresponding target properties:
   - StudentID
   - Grade
   - LastName
   - FirstName
   - MiddleName

The Editor should now look like:

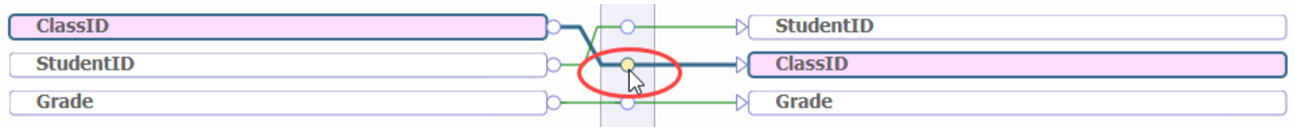


Notice that you have changed the order in which the data will be stored simply by connecting the property pairs.

2.3.2 Modifying the Value of a Property

In some cases, you want to change the value of the source's property before copying it to the corresponding target property. In this demo, the target system accepts a ClassID that starts with a two-character identifier that indicates the Source's college. The source's message does not include this identifier, so the data transformation must modify the value of ClassID before copying it to the target message.

1. Click the circle on the line that connects the source's ClassID property and the target's ClassID property.



Notice that the **Action** tab on the right-hand side of the Editor now shows information about the Assign action that was created when you connected the two ClassID properties. The **Property** field is set to `target.ClassID` and the **Value** field is set to `source.ClassID`.

2. Change the **Value** field of the **Action** tab to: `"UC."_source.ClassID`. `UC` is the college identifier and it is added to the beginning of the ClassID using the underscore, which is the concatenate operator in the InterSystems ObjectScript programming language. It is beyond the scope of this demo, but it gives you an idea of how you can include ObjectScript code in your transformation.

2.3.3 Using Functions

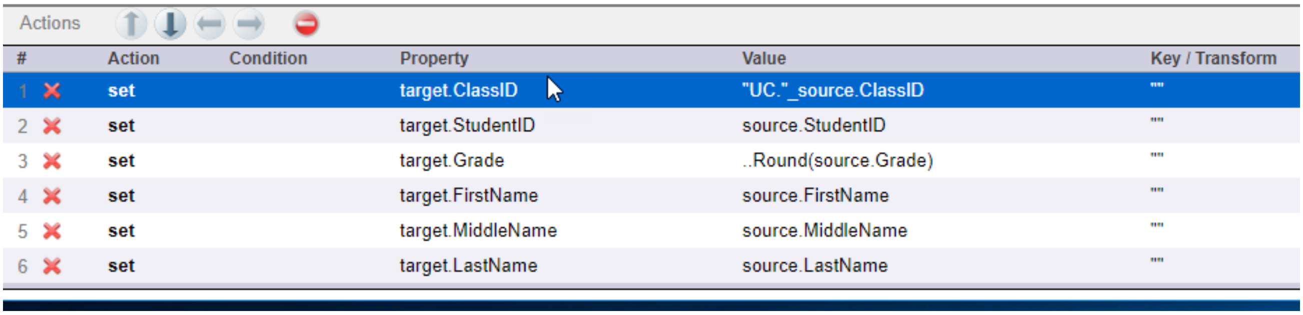
You can use built-in functions when setting the value of a property. In this demo, you are rounding the final grade to a whole number using the `Round( )` function.

1. Select the connector between the source's Grade property and the target's Grade property.
2. Click the magnifying glass next to the **Value** field of the **Action** tab.
3. Select **Round ( )** from the **Function** drop-down list.
4. Click **OK**.

The **Value** field of the **Action** tab is now set to: `..Round(source.Grade)`

2.3.4 Using the Table of Actions

Notice that as you mapped source properties to target properties, the DTL Editor added **Set** actions to the table below the diagram.



Actions

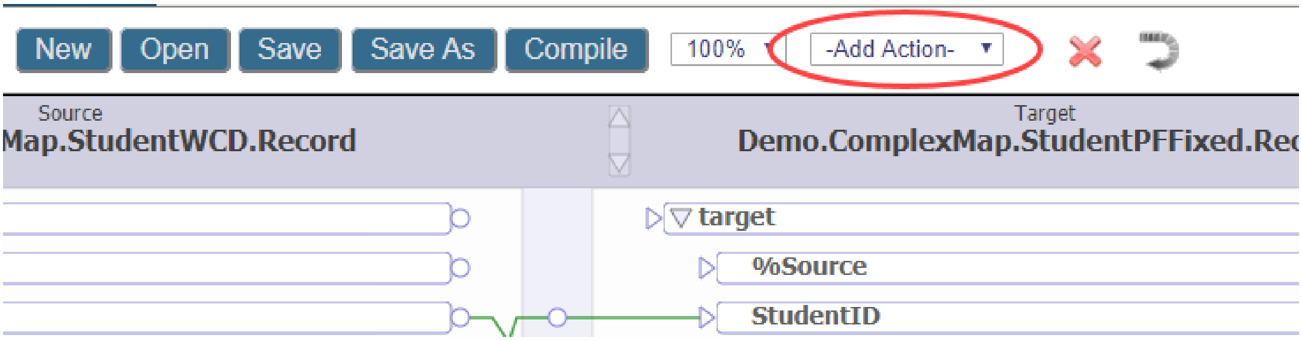
#		Action	Condition	Property	Value	Key / Transform
1	✖	set		target.ClassID	"UC."_source.ClassID	""
2	✖	set		target.StudentID	source.StudentID	""
3	✖	set		target.Grade	..Round(source.Grade)	""
4	✖	set		target.FirstName	source.FirstName	""
5	✖	set		target.MiddleName	source.MiddleName	""
6	✖	set		target.LastName	source.LastName	""

While the graphical portion of the DTL Editor makes it easy to visualize and perform actions quickly, sometimes it is better to use the table below the diagram to further define the transformation. For example, in this demo selecting the connection between the source property and target property by clicking the graphical connection was the quickest way to select the **Set** action. However, in a more complex transformation, it might be difficult to select the connector. In this case, it is easier to select the **Set** action from the table below the diagram.

2.3.5 Defining a Conditional Set of Actions

In transforming data, you can use conditional statements to perform actions only if certain conditions are true. In this demo, you define a transformation to indicate that the student passed the class if the final grade is greater than or equal to 65. To add an **If** statement to accomplish this:

1. Make sure the last item is the table is highlighted. The **If** statement is placed right after the highlighted action.
2. Select **If** from the **Add Action** drop-down list.

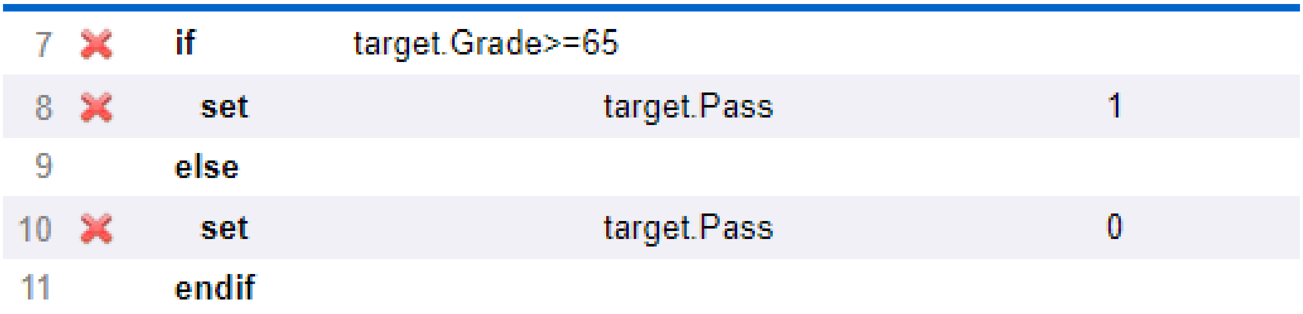


Notice that an **If** block is added to the table below the diagram.

3. With the **If** statement selected in the table, define the **Conditional** field on the **Action** tab as: `target.Grade>=65`. The actions that you put inside the **If** block are executed only if the value of the target's Grade property is greater than or equal to 65.
4. Select **Set** from the **Add Action** drop-down list.
5. On the **Action** tab, define the **Property** field as: `target.Pass`.
6. Define the **Value** field as `1`. This is the boolean value indicating that the student passed the class.
7. Select the **else** action in the table.
8. Select **Set** from the **Add Action** drop-down list.
9. On the **Action** tab, define the **Property** field as: `target.Pass`.

10. On the **Action** tab, define the **Value** field as `0`. This is the boolean value indicating that the student did not pass the class.

The **if** block should now look like:



```
if        target.Grade>=65
  set                 target.Pass        1
else
  set                 target.Pass        0
endif
```

2.3.6 Ignoring Data from Source Message

In the demo, notice that the Email, Phone, and Phone1 properties were not connected to the target message. No further action is needed to prevent this data in the source message from being copied to the target message.

2.4 Compiling the Data Transformation

You have the option to save your work as you go, but you must always remember to compile the data transformation. Your production does not recognize changes until you click **Compile**.

2.5 Performing Preliminary Testing

The DTL Editor gives you the ability to quickly test the data transformation without having to run a message through the entire production. You simply input the test message and the DTL Editor displays what the output message will look like. At this point in this demo, you are pasting an XML representation of the record map data into the Test Transform tool. Later, you will test the transformation by running a text file through the production.

Remember to compile your data transformation before testing it.

1. Click the **Tools** tab on the right-hand side of the editor.

2. Click **Test**.

3. Copy and paste the following data into the **Input Message** text box:

```
<test>
<Record>
<ClassID>CS241</ClassID>
<StudentID>930698</StudentID>
<Grade>77.8</Grade>
<LastName>Sutherland</LastName>
<FirstName>David</FirstName>
<MiddleName>Timothy</MiddleName>
<Email>david.sutherland@mail.com</Email>
<Phone>978-343-3940</Phone>
<Phone1>978-343-0951</Phone1>
</Record>
</test>
```

4. Click **Test**.

   The **Output Message** text box should now look like:

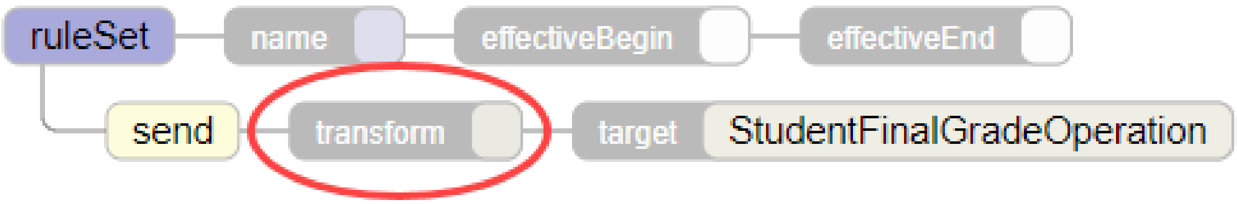
```
<Record>
  <StudentID>930698</StudentID>
  <ClassID>UC.CS241</ClassID>
  <Grade>78</Grade>
  <Pass>true</Pass>
  <FirstName>David</FirstName>
  <MiddleName>Timothy</MiddleName>
  <LastName>Sutherland</LastName>
</Record>
```

Tip: When working with HL7 or other EDI messages, you can copy and paste raw text from a sample message to test the transformation. You do not need an XML representation of the message.

2.6 Adding Transformation to Production

Now that you have created and compiled a data transformation, you must add it to your production so that source messages flowing through the production get transformed into the message that gets sent to the downstream application. In this demo, you are adding the transformation to a business rule associated with the business process DTLDemoRouter.

1. From the Management Portal home page, go to **Interoperability > List > Productions**.
2. Select `Demo.ComplexMap.SemesterProduction` and click **Open**.
3. Select `DTLDemoRouter` listed under **Processes**.
4. On the **Settings** tab, click the magnifying glass next to the **Business Rule Name** field.
5. Double-click the **transform** shape located in the business rule's **send** action.



6. In the **Data Transform Selector** dialog, select the data transformation that you created for this demo.
7. Click **OK**.
8. In the Business Rule Editor, click **Save**.

For more information about business rules, see *Developing Business Rules*.

2.7 Testing the Data Transformation

Now that you have added the data transformation to the production, run the production to see your demo in action. In this step:

- Create the directories where the production accepts and sends the sample files.
- Start the production.
- Copy the sample input file into the correct directory.
- Compare the contents of target file with the source file.

2.7.1 Creating Required Directories

The demo production is defined to look in c:\practice\in for the file containing the source data and is defined to write out the target file to c:\practice\out. You must create these directories before starting the production.

If you cannot create directories on the c:\ drive or are using a different operating system, you must modify the business service and business operation of the production before continuing. The **File Path** setting of the business service defines where the production looks for the input file while the **File Path** setting of the business operation defines where the target file is sent.

2.7.2 Starting the Production

To start the production containing the demo data transformation:

1. From the Management Portal home page, go to **Interoperability > List > Productions**.
2. Select `Demo.ComplexMap.SemesterProduction` and click **Open**.
3. Click **Start**.

2.7.3 Copying csv File to Directory

The production is defined to accept a csv file containing the student information from c:\practice\in. Copy the input.csv file from the directory where you downloaded files from GitHub into c:\practice\in. If you modified the production to specify a different directory for the business service, put input.csv into the directory that you specified.

As long as the production is running, this file should disappear from the directory within a few seconds, indicating that the production has started to process the file successfully.

2.7.4 Verifying Data in Output

The demo production takes the csv file you copied into the production's input directory and uses the data transformation to convert the data into a different format with new content. This outgoing file is copied into the production's output directory with a timestamp as the filename. By default, the production's output directory is c:\practice\out.

Use a text editor to open input.csv and the output file to compare the two.

3 Other Important Features

This demo provides the basics of developing and testing a data transformation. Other important, commonly used features include:

- **for each** statements that iterate through a collection, stepping through the same sequence of actions for every member of the collection. This collection can be an array or, in the case of HL7 or other EDI messages, a repeating segment. For more information, see Adding a For Each Action.
- Subtransformations, which allow you to modularize your data transformations. A data transformation can use a sub-transform action to run another transformation at any point in the flow of actions. You can create a library of reusable transformations that are called from other transformations, avoiding duplicate transformation logic. For example, you can create segment transformations that are reused whenever those segments appear in an EDI message.
- Support for X12 and EDIFACT standards. When creating a new transformation using the Data Transformation Wizard, you can specify whether the message type is X12 or EDIFACT. This allows you to select a document type that populates the DTL Editor with the right schema. Visual representations of the segments of these schemas expand and collapse in the DTL Editor.

4 Learn More About Data Transformation

For more details about data transformations, see:

- Developing DTL Transformations
- Advanced Data Transformations (online course)
- Business Process and Data Transformation Language Reference