# Routing EDIFACT Documents in Productions

Version 2023.3
2024-05-16

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 EDIFACT

This page briefly introduces the EDIFACT standard and InterSystems IRIS® data platform support for EDIFACT.

The Electronic Data Interchange For Administration, Commerce, and Transport standard (EDIFACT) is also known as the International Standards Organization (ISO) Standard ISO 9735-6.

United Nations rules for Electronic Data Interchange For Administration, Commerce and Transport comprise a set of internationally agreed standards, directories, and guidelines for the electronic interchange of structured data, and in particular that relate to trade in goods and services between independent, computerized information systems.

## 1.1 InterSystems IRIS Support for EDIFACT Documents

InterSystems IRIS supports EDIFACT documents as virtual documents. A *virtual document* is a kind of message that InterSystems IRIS parses only partially. This kind of message has the standard production message header and the standard message properties such as ID, Priority, and SessionId. The data in the message, however, is not available as message properties; instead it is stored directly in an internal-use global, for greater processing speed.

InterSystems IRIS provides tools so that you can access values in virtual documents, for use in data transformations, business rules, and searching and filtering messages.

## 1.2 See Also

- Using Virtual Documents in Productions

- Available Tools for Working with EDIFACT

- Configuring the Production

- Additional Steps

- Reference for EDIFACT Settings

# 2

# Available Tools for Working with EDIFACT

This page provides an overview of the InterSystems IRIS® tools that you can use to work with EDIFACT schemas and documents.

## 2.1 Using the EDIFACT Schema Structures Page

The **Interoperability** > **Interoperate** > **UN/EDIFACT** > **UN/EDIFACT Schema Structures** page enables you to import and view EDIFACT schema specifications.

On this page, you can do the following:

- View the EDIFACT schemas that have been loaded into this namespace. To do so, click **View All Schemas**.

- Import EDIFACT schemas into InterSystems IRIS. To do so, click **Import Schema from File**. Then use **Browse** to choose a file and click **OK**.

- Remove a saved EDIFACT schema. To do so, click **Remove Schema**. Then select a schema category from the **Choose Schema Category** drop-down list, and click **OK**.

  The schema is immediately removed.

  **CAUTION:**    You cannot undo the **Remove** operation.

The table on this page shows the following information:

- In each row, the two columns at far left uniquely identify the schema definition in this row:

  - **Category** — The schema category

  - **Name** — The document structure

  Table rows are sorted according to the numbers and characters in the **Category** column.

  Each value in the **Name** column is a link. When clicked, this link displays a table that outlines the EDIFACT document structure in this schema definition, showing all of its segments and fields. You can click on any of the links in this display to drill down for more details about any item. InterSystems IRIS extracts these details from the .SETS, .SEGS, .COMS, .ELMS, and .CODES sections of the SEF file that you import to define the structure of this EDIFACT document.

- **Base** — For a custom EDIFACT document structure, this column identifies the standard EDIFACT document structure on which this custom structure is based.

- **Description** — A title that describes the contents of the EDIFACT document. This **Description** is not a string that you enter as a comment or annotation. InterSystems IRIS extracts the text from the .INI section of the SEF file that you import to define the structure of this EDIFACT document.

Also see Importing EDIFACT Schemas Programmatically.

For information on creating custom schema categories, see Creating Custom Schema Categories.

## 2.2 Using the EDIFACT Document Viewer Page

The **Interoperability** > **Interoperate** > **UN/EDIFACT** > **UN/EDIFACT Document Viewer** page enables you to display EDIFACT documents, parsing them in different ways, so that you can determine which DocType to use. You can also test transformations. The documents can be external files or documents from the production message archives.

For general information on using this page, see Using the Document Viewer Page.

## 2.3 Importing SEF Files Programmatically

You can load SEF files programmatically as follows:

1. Start a Terminal session.

2. Change to an interoperability-enabled namespace and issue the following command:

   ```
   Do ##class(EnsLib.EDI.SEF.Compiler).Import(filename,"EDIFACT")
   ```

   Where *filename* is the full pathname of the SEF file and "EDIFACT" is needed to override the class default value for *filetype*.

   This command imports the data from the SEF file and makes it available as a schema definition in InterSystems IRIS.

3. InterSystems IRIS creates a name for the new schema category from the first piece of the first line in the .INI section of the SEF file. For example, in the file D96A.sef you might see this:

   ```
   .INI
   D96A,,D 96A,UN,D96A,D96A schema
   ```

   The extracted schema category has the name D96A.

   Due to the schema naming convention, if you want to edit a SEF file to customize it, InterSystems suggests you first change the text in the SEF file that provides its category name, so that you can distinguish your version from any other SEF file that you also import into InterSystems IRIS.

4. A SEF file may contain syntax errors. If so, InterSystems IRIS issues an error message and identifies the location of the error in the SEF file.

## 2.4 EDIFACT Classes

For reference, this section lists the classes that InterSystems IRIS provides to enable you to work with EDIFACT documents.

| Item | Classes | Notes |
|------|---------|-------|
| Business services | • EnsLib.EDI.EDIFACT.Service.FileService<br>• EnsLib.EDI.EDIFACT.Service.FTPService<br>• EnsLib.EDI.EDIFACT.Service.HTTPService | Each of these EDIFACT business service classes uses a different adapter, as indicated by the class name. |
| Business processes | EnsLib.MsgRouter.VDocRoutingEngine | This class is the standard virtual document routing process. |
| Business operations | • EnsLib.EDI.EDIFACT.Operation.FileOperation<br>• EnsLib.EDI.EDIFACT.Operation.FTPOperation<br>• EnsLib.EDI.EDIFACT.Operation.HTTPOperation | Each of these EDIFACT business operation classes uses a different adapter, as indicated by the class name. |
| Messages | EnsLib.EDI.EDIFACT.Document | This is a specialized message class to carry EDIFACT documents as virtual document. |
| Search tables | EnsLib.EDI.EDIFACT.SearchTable | This is a specialized search table class for EDIFACT documents. |

You can also create and use subclasses of these classes.

The business host classes include configurable targets. The following diagram shows some of them:
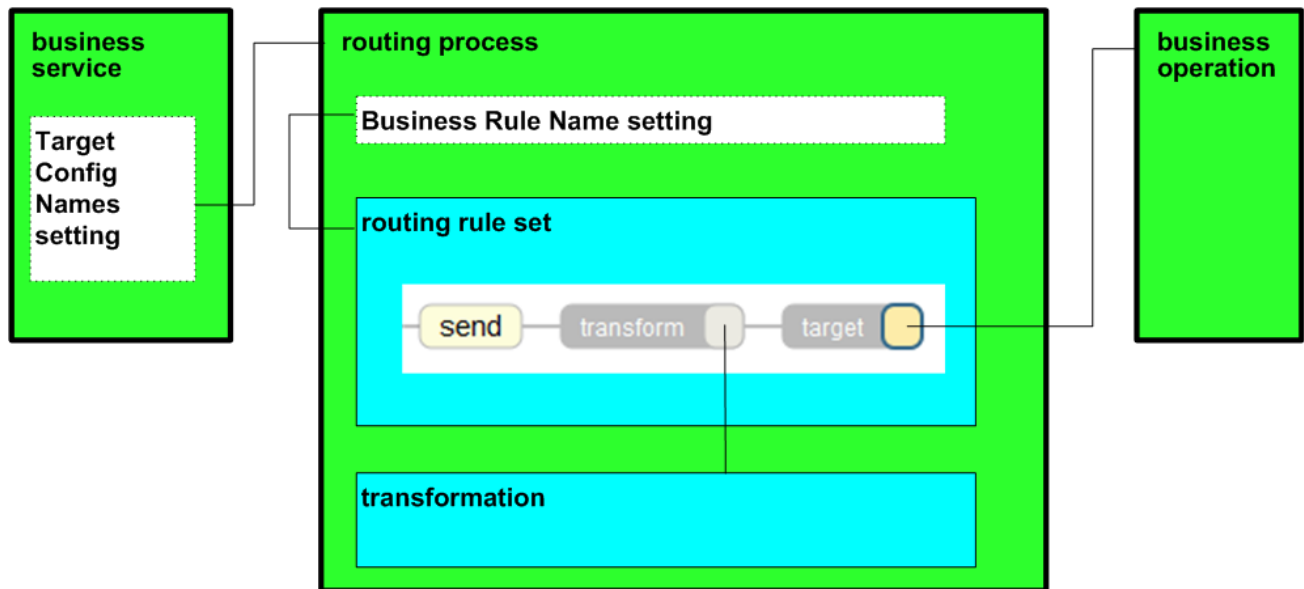


For information on other configurable targets, see Reference for Settings.

# 2.5 See Also

- Introduction to EDIFACT

- Using Virtual Documents in Productions

- [Configuring the Production](#)
- [Additional Steps](#)
- [Reference for EDIFACT Settings](#)

# 3

# Configuring the Production (EDIFACT)

This page describes how to configure a production to include an EDIFACT routing interface. It discusses tasks that you perform on the **Interoperability** > **Configure** > **Production** page. The following article describes additional tasks.

## 3.1 Adding EDIFACT Business Services

Add one EDIFACT business service for each document type that the production will receive. If this document type arrives via multiple communication modes (for example, via FTP in addition to TCP), you will need a business service for each communication mode.

To add an EDIFACT business service to a production, use the Business Service Wizard as usual; see Configuring Productions. Select one of the following classes from the **Service Class** list:

- EnsLib.EDI.EDIFACT.Service.FileService

- EnsLib.EDI.EDIFACT.Service.FTPService

- EnsLib.EDI.EDIFACT.Service.HTTPService

## 3.2 Adding EDIFACT Routing Processes

To add a routing process to a production, use the Business Process Wizard as usual; see Configuring Productions. Select EnsLib.MsgRouter.VDocRoutingEngine from the **Process Class** list.

## 3.3 Adding EDIFACT Business Operations

Add an EDIFACT business operation for each output destination.

You might also want to add business operations to handle bad messages (for background, see Business Processes for Virtual Documents).

To add an EDIFACT business operation to a production, use the Business Operation Wizard as usual; see Configuring Productions. Select one of the following classes from the **Operation Class** list:

- EnsLib.EDI.EDIFACT.Operation.FileOperation

- EnsLib.EDI.EDIFACT.Operation.FTPOperation

- EnsLib.EDI.EDIFACT.Operation.HTTPOperation

# 3.4 Connecting the EDIFACT Business Hosts

After you add the EDIFACT business hosts, connect these items as follows:

- For each EDIFACT business service, specify the Target Config Names setting as the name of the EDIFACT routing process.

- Create a routing rule set that contains the desired logic. See the next article.

  For the routing rule set, make sure that the **Target** field is the EDIFACT business operation.

- For the EDIFACT routing process, specify the Business Rule Name setting. Use the full name of the new routing rule set.

# 3.5 Configuring the Business Hosts

You should examine all the settings listed in Reference for Settings and set them as needed.

A couple of key settings for an EDIFACT business service are as follows:

- Doc Schema Category — Specifies the schema category to assign to the inbound documents. InterSystems IRIS® requires this information for validation and for search table indexing.

- Search Table Class — Specifies the class to use to index virtual properties in the inbound documents.

- For a File or FTP business service, also consider whether you need to configure the Reply Target Config Names field.

Also be sure to configure the Separators setting.

# 3.6 See Also

- Introduction to EDIFACT

- Using Virtual Documents in Productions

- Available Tools for Working with EDIFACT

- Configuring Productions

- Additional Steps

- Reference for EDIFACT Settings

# 4

# Additional Steps (EDIFACT)

This page discusses the additional steps needed to add EDIFACT routing interfaces to a production. Be sure to perform these tasks in the same namespace that contains your production. When you create rule sets, transformations, and search tables, do not use reserved package names; see Reserved Package Names.

Also see Overriding the Validation Logic.

## 4.1 Loading EDIFACT Schemas into InterSystems IRIS

To load an EDIFACT schema into InterSystems IRIS®, use the EDIFACT Schema Structures page, described in Available Tools.

For information on creating custom schema categories, see Creating Custom Schema Categories.

## 4.2 Defining Routing Rule Sets for EDIFACT

For general information on defining business rules, see *Developing Business Rules*.

When you create a routing rule set for an EDIFACT routing interface:

- On the **general** tab, **Rule Type** should be **Virtual Document Message Routing Rule**. This choice sets the following options:

    - **Rule Assist Class** should be EnsLib.MsgRouter.VDocRuleAssist

    - **Context Class** should be EnsLib.MsgRouter.VDocRouting Engine

- In the **constraint** for a rule, specify **Message Class** as EnsLib.EDI.EDIFACT.Document.

## 4.3 Defining DTL Data Transformations for EDIFACT

Your routing rules might need one or more data transformations.

For general information on defining DTL data transformations, see *Developing DTL Transformations*.

When you create a DTL data transformation for EDIFACT documents:

- On the **Transform** tab, **Source Class** and **Target Class** should both be EnsLib.EDI.EDIFACT.Document.

- **Source Doc Type** should match the schema category name assigned by the business service.

- **Target Doc Type** should be the name of the target schema category. This must match a schema category name that you have loaded into InterSystems IRIS.

Use the EDIFACT Document Viewer Page to test your transformations, as described in Available Tools.

To integrate the DTL data transformation in the production, enter its full package and class name in the **Transform** field of a routing rule set.

# 4.4 Defining EDIFACT Search Tables

The EDIFACT search table class, EnsLib.EDI.EDIFACT.SearchTable, automatically indexes the EDIFACT document ID, which it gives the name Identifier.

If you need more items to search, you can create a subclass. The subclass inherits the Identifier property, plus the infrastructure that makes search tables work. For details, see Defining a Search Table Class.

Note the following points specific to EDIFACT:

- See the list of EDIFACT separator characters, in the reference for the Separator setting.

- In this case, InterSystems IRIS supports an additional value for PropType. You can use DateTime:HL7 in addition to the types listed in Using Virtual Documents in Productions.

# 4.5 Handling Repetitions in EDIFACT Documents

Some segments in EDIFACT documents can repeat or be used within loops. The repeat and loop structure can be either expressed implicitly or explicitly. InterSystems IRIS can handle either implicit or explicit repeat and loop structures, but the document must either specify all repeats and loops explicitly or all implicitly. When InterSystems IRIS is parsing an EDIFACT document and encounters the first repetition or loop, it determines whether the document is using implicit or explicit repeats. It then parses the remainder of the document using the mechanism found in the first repetition or loop. When it is parsing an EDIFACT document, InterSystems IRIS treats a mixture of explicit and implicit repeat mechanisms as an error.

If an EDIFACT document is using implicit indication of repetition, a segment is represented as a three letter label (such as UNH), followed by the data element separator and data (the data elements and components). No information explicitly indicates which repetition of a segment or a loop a particular segment is in. But if an EDIFACT document is using explicit indication of repetition, a repeating segment is expressed as the three letter label, followed by the component separator and the control numbers, followed by the data element separator. For a segment, ARA, which is a repeating segment, this may look like ARA:1+data. For a segment, DET, which is in Group 2, which is nested in Group 1, this may look like DET:1:1+data. Only those segments which can repeat or which are inside a loop are expressed using these control numbers.

By default, an EDIFACT document retains its implicit or explicit repetition indication when output to a file. You can convert an EDIFACT document with implicit indication to explicit by using the **ConstructExplicitClone** method, and you can convert an EDIFACT document with explicit indication to implicit by using the **ConstructExplicitClone** method.

# 4.6 See Also

- Using Virtual Documents in Productions
- Creating Custom Schema Categories
- Developing Business Rules
- Defining a Search Table Class
- Introduction to EDIFACT
- Available Tools for Working with EDIFACT
- Configuring the Production
- Reference for EDIFACT Settings

# Reference for EDIFACT Settings

This section provides the reference information for settings for EDIFACT business services and EDIFACT business operations.

For information on settings for the routing process (EnsLib.MsgRouter.VDocRoutingEngine), see Settings of a Virtual Document Routing Process.

# Settings for EDIFACT Business Services

Provides reference information for settings of EDIFACT business services.

## Summary

EDIFACT business services provide the following settings:

| Group | Settings |
|---|---|
| Basic Settings | Target Config Names, Doc Schema Category |
| Additional Settings | Reply Target Config Names, Search Table Class, Validation, Reply Mode, Batch Handling, Local Application ID, Tolerate Newlines |

The remaining settings are either common to all business services or are determined by the type of adapter. For information, see:

- Settings for All Business Services

- Settings for the File Inbound Adapter

- Settings for the FTP Inbound Adapter

- Settings for the HTTP Inbound Adapter

## Batch Handling

Indicates how to treat received batch Interchange documents the production receives. The options are:

- `Whole Batch` — Do not process child documents individually; accumulate and send the whole batch as one composite document.

- `Single-Session Batch` — Forward each document in the batch as part of a single session, including a final parent document object containing the batch header and trailer segments.

- `Multi-Session Batch` — Forward each document in the batch in its own session, followed by the parent document object containing the batch header and trailer segments.

- `Individual` — Forward each child document in the batch in its own session; do not forward parent batch document objects.

## Doc Schema Category

Category to apply to incoming EDIFACT document type names to produce a complete DocType specification. Combines with the document type name to produce a DocType assignment. This setting may also contain multiple comma-separated type names followed by = and a DocTypeCategory, or full DocType values to apply to documents declared as that type.

A trailing asterisk (*) at the end of a given partial type name matches any types beginning with the partial entry.

For example:

```
D96A,REC*=D04A,REQOTE=D05B
```

Note that a DocType assignment may be needed for **Validation** or Search Table Class indexing.

## Local Application ID

Colon-separated *LocalID*:*Qualifier* code that represents the facility and application that receive EDIFACT documents via this business service. These are used to create reply document headers. The @ (at sign) character represents using the corre-

sponding field from the incoming message. If your ID must contain a literal @ symbol, escape it with back slash as follows: \@. The default value is: `EDIFACTService:ENS`

## Reply Mode

Controls response handling and whether or not to send back reply documents immediately upon receipt on an interchange. The **Reply Mode** options are:

* `Never` — Do not send back any immediate reply. This is the default.

* `All` — Generate a reply for every transaction set in an interchange.

* `Errors` — Only generate a reply for transaction sets in which errors are detected.

* `Success` — Only generate a reply for transaction sets that are accepted without errors.

## Reply Target Config Names

(File and FTP only) Comma-separated list of configuration items within the production to which the business service should relay any EDIFACT *reply* messages. Usually the list contains one item, but it can be longer. The list can include both business processes and business operations.

Compare to **Target Config Names**.

## Search Table Class

Specifies the class to use to index virtual properties in the inbound documents. The default is EnsLib.EDI.EDIFACT.SearchTable. To use a different class, see Defining EDIFACT Search Tables.

In either case, be sure that the category given by Doc Schema Category includes the DocType values (if any) in the search table class.

## Target Config Names

Comma-separated list of configuration items to which to send EDIFACT documents. The value of this field must be the configured name of one or more of the following items within the production:

* A routing process (for a routing interface)

* A business operation (if your design bypasses a routing process for this interface and simply relays documents from the incoming business service to the outgoing business operation)

Compare to **Reply Target Config Names**.

## Tolerate Newlines

True or False. If True, the business service processes an incoming file without error, even if newline characters have been inserted into the file after (or in place of) segment terminators to enhance readability. If False, these extra newline characters trigger an error in parsing the file. The default is True.

## Validation

See Validation in Settings of a Virtual Document Business Service.

Also see the reference for the Separator setting.

# Settings for EDIFACT Business Operations

Provides reference information for settings of EDIFACT business operations.

## Summary

EDIFACT business operations provide the following settings:

| Group | Settings |
|---|---|
| Basic Settings | File Name |
| Additional Settings | Auto Batch Parent Segs, Separators, Search Table Class, Validation, Reply Code Actions, No Fail While Disconnected |

The remaining settings are either common to all business operations or are determined by the type of adapter. For information, see:

*   Settings for All Business Operations

*   Settings for the File Outbound Adapter

*   Settings for the FTP Outbound Adapter

*   Settings for the HTTP Outbound Adapter

## Auto Batch Parent Segs

(File and FTP only) If set to True, when writing a message that has a batch parent, output the batch headers first, and follow up with the batch trailers when triggered by the final batch parent header message or by a file name change. All child messages of a batch parent message are written out unless already written previously while **Auto Batch Parent Segs** = True.

## File Name

(File and FTP only) The target file name. The **File Path** adapter setting determines the path for this file; **File Name** determines the name. **File Name** can include time stamp specifiers. If you leave **File Name** blank, the default uses the time stamp specifier `%f_%Q` where:

*   `%f` is the name of the data source, in this case the input filename

*   `_` is the literal underscore character, which will appear in the output filename

*   `%Q` indicates ODBC format date and time

In substituting a value for the format code `%f`, InterSystems IRIS strips out any of the characters `|,?,\,/,:,[,],<,>,&,,,;,NUL,BEL,TAB,CR,LF`, replacing spaces with underscores (`_`), slashes (`/`) with hyphens (`-`), and colons (`:`) with dots (`.`).

See Time Stamp Specifications for Filenames.

## No Fail While Disconnected

(HTTP only) If set to True, suspend counting seconds toward **Failure Timeout** while disconnected. Does not apply if **Failure Timeout** = `-1` or **StayConnected** = `0`.

## Reply Code Actions

(HTTP only) A comma-separated list of codes specifying what action this process takes on various reply status conditions. See Reply Code Actions.

HTTP adapters provide more additional possible values; see EnsLib.EDI.EDIFACT.Operation.ReplyStandard in the *Class Reference*.

## Search Table Class

See Search Table Class in Settings for EDIFACT Business Services.

## Separators

A string specifying characters to be used as separators for encoding outbound EDIFACT documents.

**Separators** accepts up to seven characters as input and defaults to :+?*'\r\n. Each character of input is interpreted from left to right as follows:

| Position | Separator | Default value |
|----------|-----------|---------------|
| 1 | Component Separator | : (colon) |
| 2 | Data Element Separator | + (plus sign) |
| 3 | Escape Separator | ? (question mark) |
| 4 | Repetition Separator | * (asterisk) |
| 5 | Segment Terminator | ' (apostrophe) |
| 6–7 | appendix to Segment Terminator | \r\n (new line) |

If the **Separators** string is empty, the separator values supplied within each EDIFACT document will be used.

Any characters in positions 6 and 7 will be appended to the *Segment Terminator*. By default \r\n occupy these positions where \r (ASCII 13) represents a carriage return and \n (ASCII 10) represents a line feed. These characters resolve to a new line such that each segment is represented on its own line in output.

**Note:** A question mark (?) also acts as an escape for characters reserved as separators. InterSystems IRIS will automatically escape any reserved characters in an EDIFACT message that were not previously separators. Conversely, InterSystems IRIS will un-escape any characters that were previously reserved but are no longer. For example, if ^ was the default *Component Separator* in a document and you set : to override it then any : characters will be escaped as ?:, any ^ separators will be converted to :, and any escaped ?^ characters will be converted to ^.

## Validation

String specifying types of validation to perform; see EnsLib.EDI.EDIFACT.Validator in the *Class Reference* for details. Any nonzero string invokes basic validation.