Information Builders

iWay

iWay Integration Solution for SWIFT 2019 User's Guide

Version 8.0 and Higher

July 29, 2019

Contents

Preface

This documentation describes how to configure and use the iWay Integration Solution for SWIFT. It is intended for developers to enable them to parse, transform, validate, store, and integrate information into the existing enterprise and pass information electronically to partners in Society for Worldwide Interbank Financial Telecommunication (SWIFT) mandated format.

How This Manual Is Organized

This manual includes the following chapters:

	Chapter/Appendix	Contents
1	Introducing the iWay Integration Solution for SWIFT	Describes the Society for Worldwide Interbank Financial Telecommunication (SWIFT) and how the components of the iWay Integration Solution for SWIFT streamline the flow of information.
2	Deployment Information for Your iWay Integration Solution	Describes the iWay products used with your iWay Integration Solution for SWIFT and provides a roadmap to full information on those products. Introduces the concept of a channel for the construction of a message flow in iWay Service Manager.
3	Configuring the EDI Activity Driver	Describes how to configure the EDI Activity Driver using iWay Service Manager.
4	Working With SWIFT Inbound and Outbound Sample Applications Using iWay Integration Tools	Describes how to work with SWIFT inbound and outbound sample applications using iWay Integration Tools (iIT).
5	Inbound Processing: SWIFT to XML	Includes an overview of the iWay business components and processing steps in a basic inbound message flow. The message flow converts a SWIFT FIN formatted message to an XML transaction. Also includes instructions for configuring a basic inbound message flow.

	Chapter/Appendix	Contents
6	Outbound Processing: XML to SWIFT	Includes an overview of the iWay business components and processing steps in a basic outbound message flow. The message flow converts a document from XML format to SWIFT format. Also includes instructions for configuring a basic outbound message flow.
A	Ebix-Supported Transaction Sets	Describes the SWIFT FIN messages supported by the iWay Integration Solution for SWIFT in the Ebix files supplied with the product.
B	Using iWay Integration Tools to Configure an Ebix for SWIFT	Describes how to use iWay Integration Tools (iIT) to configure an e-Business Information Exchange (Ebix) file for SWIFT.
C	Sample SWIFT Documents	Includes sample SWIFT MT950 (Statement of Cash), MT535 (Statement of Holding), and MT541 (Receive Against Payment) documents.

Documentation Conventions

The following table describes the documentation conventions that are used in this manual.

Convention	Description
THIS TYPEFACE or this typeface	Denotes syntax that you must enter exactly as shown.
this typeface	Represents a placeholder (or variable), a cross-reference, or an important term. It may also indicate a button, menu item, or dialog box option that you can click or select.
underscore	Indicates a default setting.
Key + Key	Indicates keys that you must press simultaneously.
{ }	Indicates two or three choices. Type one of them, not the braces.
|	Separates mutually exclusive choices in syntax. Type one of them, not the symbol.

Convention	Description
...	Indicates that you can enter a parameter multiple times. Type only the parameter, not the ellipsis (...).
. . .	Indicates that there are (or could be) intervening or additional commands.

Related Publications

Visit our Technical Documentation Library at *http://documentation.informationbuilders.com*. You can also contact the Publications Order Department at (800) 969-4636.

Customer Support

Do you have any questions about this product?

Join the Focal Point community. Focal Point is our online developer center and more than a message board. It is an interactive network of more than 3,000 developers from almost every profession and industry, collaborating on solutions and sharing tips and techniques. Access Focal Point at *http://forums.informationbuilders.com/eve/forums*.

You can also access support services electronically, 24 hours a day, with InfoResponse Online. InfoResponse Online is accessible through our website, *http://www.informationbuilders.com*. It connects you to the tracking system and known-problem database at the Information Builders support center. Registered users can open, update, and view the status of cases in the tracking system and read descriptions of reported software issues. New users can register immediately for this service. The technical support section of *http://www.informationbuilders.com* also provides usage techniques, diagnostic tips, and answers to frequently asked questions.

Call Information Builders Customer Support Services (CSS) at (800) 736-6130 or (212) 736-6130. Customer Support Consultants are available Monday through Friday between 8:00 a.m. and 8:00 p.m. EST to address all your questions. Information Builders consultants can also give you general guidance regarding product capabilities and documentation. Please be ready to provide your six-digit site code number (*xxxx.xx*) when you call.

To learn about the full range of available support services, ask your Information Builders representative about InfoResponse Online, or call (800) 969-INFO.

Help Us to Serve You Better

To help our consultants answer your questions effectively, be prepared to provide specifications and sample files and to answer questions about errors and problems.

The following tables list the environment information our consultants require.

Platform	
Operating System	
OS Version	
JVM Vendor	
JVM Version	

The following table lists the deployment information our consultants require.

Adapter Deployment	For example, JCA, Business Services Provider, iWay Service Manager
Container	For example, WebSphere
Version	
Enterprise Information System (EIS) - if any	
EIS Release Level	
EIS Service Pack	
EIS Platform	

The following table lists iWay-related information needed by our consultants.

iWay Adapter	
iWay Release Level	
iWay Patch	

The following table lists additional questions to help us serve you better.

Request/Question	Error/Problem Details or Information
Did the problem arise through a service or event?	
Provide usage scenarios or summarize the application that produces the problem.	
When did the problem start?	
Can you reproduce this problem consistently?	
Describe the problem.	
Describe the steps to reproduce the problem.	
Specify the error message(s).	
Any change in the application environment: software configuration, EIS/database configuration, application, and so forth?	
Under what circumstance does the problem *not* occur?	

The following is a list of error/problem files that might be applicable.

- ❑ Input documents (XML instance, XML schema, non-XML documents)
- ❑ Transformation files
- ❑ Error screen shots
- ❑ Error output files
- ❑ Trace files

- Service Manager package to reproduce problem
- Custom functions and agents in use
- Diagnostic Zip
- Transaction log

For information on tracing, see the *iWay Service Manager User's Guide*.

User Feedback

In an effort to produce effective documentation, the Technical Content Management staff welcomes your opinions regarding this document. Please use the Reader Comments form at the end of this document to communicate your feedback to us or to suggest changes that will support improvements to our documentation. You can also contact us through our website, *http://documentation.informationbuilders.com/connections.asp*.

Thank you, in advance, for your comments.

Information Builders Consulting and Training

Interested in training? Information Builders Education Department offers a wide variety of training courses for this and other Information Builders products.

For information on course descriptions, locations, and dates, or to register for classes, visit our website (*http://education.informationbuilders.com*) or call (800) 969-INFO to speak to an Education Representative.

Chapter 1

Introducing the iWay Integration Solution for SWIFT

The iWay Integration Solution for SWIFT transforms Society for Worldwide Interbank Financial Telecommunication (SWIFT) documents into XML format, or transforms XML representations into SWIFT format.

This section provides an overview of SWIFT FIN and describes the features that are provided by the iWay Integration Solution for SWIFT.

In this chapter:

A Brief History of SWIFT

Society for Worldwide Interbank Financial Telecommunication (SWIFT) is a set of standards for formatting information that is electronically exchanged between one financial or corporate entity and another.

By specifying a standardized, computer-readable format for transferring data, SWIFT enables the automation of financial transactions around the world. It provides a common, uniform language through which computers can communicate for fast and efficient transaction processing.

Early Standardization Efforts

Society for Worldwide Interbank Financial Telecommunication (SWIFT) provides secure messaging services and interface software to more than 11,000 financial institutions in more than 212 countries.

SWIFT was founded in 1973 by 239 banks in 15 countries to create a unified international transaction processing and transmission system. In 1977, Prince Albert of Belgium sent the first SWIFT message and there were in excess of 513 member banks in more than 15 countries. In 1987, SWIFT membership expanded to include financial institutions operating in securities and money markets.

The SWIFT FIN Standard

Development of the SWIFT FIN standards have progressed since its inception. Each year members of the SWIFT community meet in order to review their current SWIFT FIN standards and present ideas relating to the addition of new transactions, modification of existing transactions and the removal of no longer needed transactions. This is all driven by the financial and corporate user community.

The International Organization for Standardization (ISO) is a worldwide federation of national standards bodies (ISO member bodies).

International Standard ISO 15022 was prepared by Technical Committee ISO/TC68, Banking, Securities and Related Financial Services, Sub-Committee SC4, Securities and Related Financial Instruments. It replaced the previous standards for electronic messages exchanged between securities industry members, ISO 7775 - Scheme for message types and ISO 11521 - Scheme for interdepository message types. The current SWIFT FIN standard is based on ISO 15022.

ISO 15022 sets the principles necessary to provide the different communities of users with the tools to design message types to support their specific information flows. These tools consist of a set of syntax and message design rules, a dictionary of data fields and a catalog for present and future messages built by the industry with the above mentioned fields and rules.

To address the evolving needs of the industry as they arise, the Data Field Dictionary and Catalog of Messages have been kept outside the standard. They are made available by the Registration Authority which maintains them as necessary upon the request of industry participants.

To protect investments already made by the industry, the syntax proposed in this standard, referred to as "Enhanced ISO 7775 syntax", is based on the syntax used for the previous ISO 7775 and ISO 11521.

ISO 15022 has been designed to incorporate and be upwards compatible with the previous securities message standards ISO 7775 and ISO 11521, as updated in ISO TR 7775. As a result, the initial Data Field Dictionary and Catalogue of Messages accommodate ISO TR7775 data fields and messages. However, some of the previous fields and messages are not fully compliant with the Enhanced ISO 7775 syntax. In addition the initial Data Field Dictionary incorporates the Industry Standardization for Institutional Trade Communications (ISITC) DSTU 1/1995 and the Securities Standards Advisory Board (SSAB) data dictionaries.

The ISO 15022 standard is described in the document "ISO 15022 Securities - Scheme for Messages (Data Field Dictionary), Part 1 - Data Field and Message Design Rules and Guidelines and Part 2 - Maintenance of the Data Field Dictionary and Catalogue of Messages".

Orders for ISO 15022 and other International Standards or ISO publications should be addressed to the ISO member bodies which are normally the primary sales agents in their country. For customers in countries where there is no member body, the order should be addressed to the ISO Central Secretariat.

Features of the iWay Integration Solution for SWIFT

The standards-based iWay Integration Solution for SWIFT reduces the amount of effort it takes to integrate SWIFT documents with your internal enterprise applications and third-party trading partners. It includes conversion and validation of documents from SWIFT to XML format, making it easy to include SWIFT documents in your XML-based integration projects.

Features of the iWay Integration Solution for SWIFT include:

- Integration with iWay Service Manager to provide bi-directional conversion of SWIFT formats and XML between application servers, integration brokers, third-party software packages, and messaging services.
- Integration with more than 200 other information assets, including J2EE-based back-office systems; data structures such as DB2, IMS, VSAM, and ADABAS; and front-office systems based on Sybase.
- Integration with leading application servers, integration brokers, and development environments. Supported software platforms include BEA WebLogic, IBM WebSphere, Sun Java Enterprise System, and Oracle Application Server.
- Out-of-the-box support for SWIFT FIN messages. For details on the supported messages, see *Ebix-Supported Transaction Sets* on page 89.
- Reusable framework for parsing, transforming, and validating SWIFT documents without the need to write custom code.

Information Roadmap

The following table lists the location of deployment and user information for products used with the iWay Integration Solution for SWIFT.

Product	For more information, see...
iWay Service Manager	Chapters 3, 4, and 5 of this guide *iWay Service Manager User's Guide*

Product	For more information, see...
iWay Integration Tools (iIT) Transformer	*iWay Integration Tools Transformer User's Guide*
iWay Integration Tools (iIT) Designer	*iWay Integration Tools Designer User's Guide*

Chapter 2

Deployment Information for Your iWay Integration Solution

This topic describes the iWay products used with your iWay Integration Solution for SWIFT and provides a roadmap to full information on those products.

It also introduces the concept of a channel for the construction of a message flow in iWay Service Manager.

In this chapter:

iWay Products

Your iWay integration solution works in conjunction with one or more of the following products:

- iWay Service Manager
- iWay Integration Tools (iIT) Transformer
- iWay Integration Tools (iIT) Designer

iWay Service Manager

iWay Service Manager is the heart of the Universal Integration Framework and is an open transport service bus. Service Manager uses graphical tools to create sophisticated integration services without writing custom integration code by:

- Using metadata from target applications
- Transforming and mapping interfaces
- Managing stateless processes

Its capability to manage complex integration interactions makes it ideally suited to be the foundation of a service-oriented architecture.

For more information, see the *IWay Service Manager User's Guide*.

iWay Integration Tools Transformer

iWay Integration Tools (iIT) Transformer is a GUI tool that is a plugin to iIT. Transformer is a rule-based data transformation tool that converts an input document of one data format to an output document of another data format or structure. The easy-to-use graphical user interface and function tool set facilitate the design of transform projects that are specific to your requirements.

For more information, see the *iWay Integration Tools Transformer User's Guide*.

iWay Integration Tools Designer

The capability of graphically visualizing a business process is a powerful and necessary component of any e-Business offering. iWay Integration Tools (iIT) Designer is a GUI tool that is a plugin to iIT. iIT Designer provides a visual and user-friendly method of creating a business process flow. Through a process flow, you control the sequence in which tasks are performed and the destination of the output from each task.

For more information, see the *iWay Integration Tools Designer User's Guide*.

Using a Channel to Construct a Message Flow

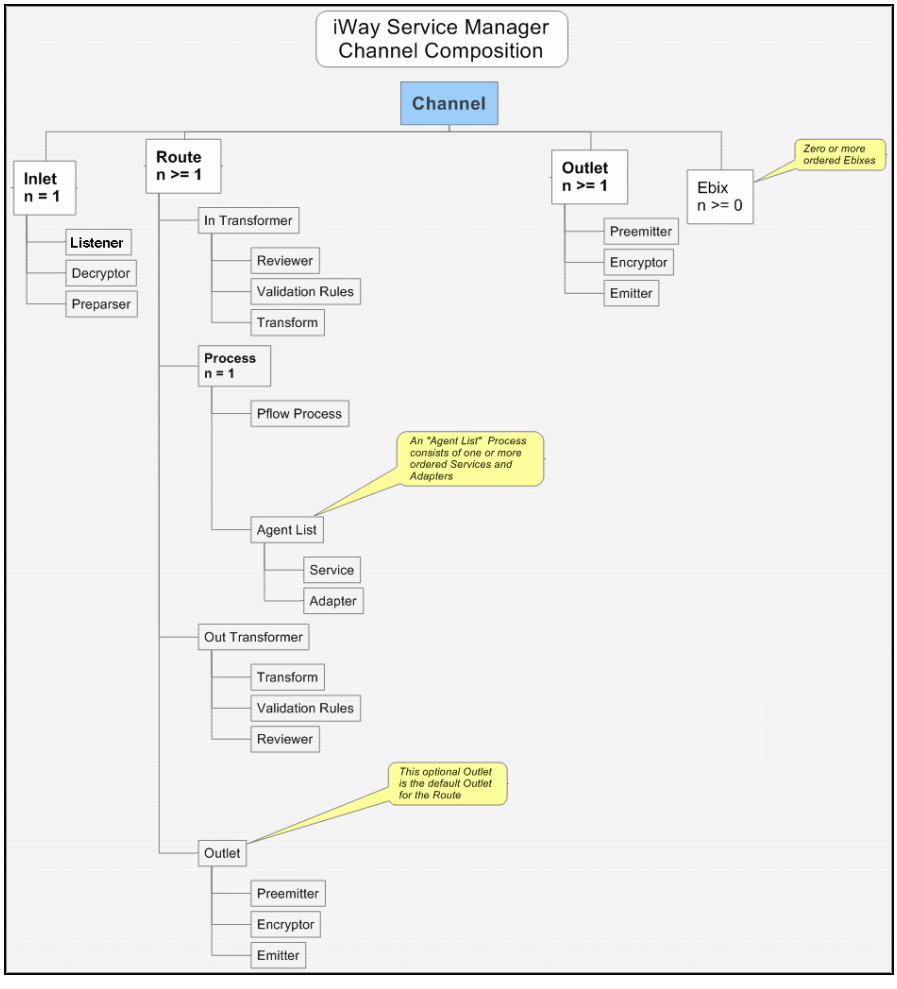
The use of iWay Service Manager is centered on a channel. A channel is a container for all the iWay business components used in an SWIFT message flow.

At a high level, a channel accepts input data via an **inlet**, processes the data via a **route**, and outputs the resulting data via an **outlet**. Another component in the process is an e-Business Information Exchange (**Ebix**).

The following diagram shows the channel components available in the construction of a message flow.

In the following diagram, the value **n** underneath a component name indicates how many instances of that component you can have in a channel configuration—zero, one, or more than one. For example, n = 1 for Inlet means that you can have only one inlet on the channel.

Required components are in boldface type.



Components of a Channel

A channel consists of:

- An inlet, which defines how a message enters a channel.
- A route, which defines the path a message takes through a channel.
- Outlets, which define how transformed messages leave a channel.
- An e-Business Information Exchange (Ebix), which is a collection of metadata that defines the structure of data.

iWay Service Manager provides a design-time repository called the Registry, where you assemble and manage the components in a channel.

An **inlet** can contain:

- A listener (required), which is a protocol handler responsible for picking up an incoming message on a channel.
- A decryptor, which applies a decryption algorithm to an incoming message and verifies the security of the message.
- A preparser, which is a logical process that converts an incoming message into a processable document. The preparsed document then passes through the standard transformation services to reach the designated processing service.

A **route** can contain:

- An *in* transformer, which is an exit sequence that is applied to a message before processing occurs.
  - A reviewer, which is either the first exit to receive a document after parsing (inbound), or the last exit to receive a document prior to the actual emit operation (outbound). These exits are intended for envelope handling but can be used for any desired purpose.
  - Validation rules, which apply validation using the rules validation engine. Rules are provided when the iWay Integration Solution for SWIFT is installed.
  - A transform, which is a transformation definition file that contains sets of rules, interpreted and executed by a transformation engine. Transformation is the process by which data is transformed from one structure/format to another.
- A process, which is a stateless, lightweight, short-lived microflow that is executed by iWay Service Manager on a message as it passes through the system. Processes that are published using iIT Designer are available in the Registry and can be bound to channels as routes.
  - A process flow.
  - An agent list.
    - A service, which is an executable Java procedure that handles the business logic of a message.

- An adapter, which refers to a target that represents a specific instance of a connection to a back-end system.

- An *out* transformer, which is an exit sequence that is applied to a message after processing occurs.
  - A transform, which is a transformation definition file that contains sets of rules, interpreted and executed by a transformation engine. Transformation is the process by which data is transformed from one structure/format to another.
  - Validation rules, which apply validation using the rules validation engine. Rules are provided when the iWay Integration Solution for SWIFT is installed.
  - A reviewer, which is either the first exit to receive a document after parsing (inbound), or the last exit to receive a document prior to the actual emit operation (outbound). These exits are intended for envelope handling but can be used for any desired purpose.
- An outlet (optional), which is responsible for all aspects of preparing a document for emission and then emitting it.
  - A preemitter, which is a logical process that handles a document immediately before transmission. Normally it converts an XML document into non-XML format.
  - An encryptor, which can be called to encrypt an outgoing document.
  - An emitter, which is a transport protocol that sends a document to its recipient.

An **outlet** can contain:

- A preemitter.
- An encryptor.
- Multiple emitters.

For details on the preceding components, see the *iWay Service Manager User's Guide*.

Components of the iWay Integration Solution for SWIFT

iWay business components used in the construction of a message flow for SWIFT messages include:

- An Ebix (e-Business Information Exchange)
- A preparser
- A validation report

- A preemitter

Ebix

iWay Software provides various e-Business Information Exchange (Ebix) files used in conjunction with the iWay integration solutions. In iWay Service Manager, the iWay Integration Solution for SWIFT contains several Ebix files, one for each supported SWIFT FIN year.

Ebix files for SWIFT FIN are named SWIFT_*ccyy*.ebx, where *ccyy* is the release year. For example, the Ebix file for the 2019 SWIFT FIN message is named SWIFT_2019.ebx.

For details on the supported SWIFT FIN messages, see *Ebix-Supported Transaction Sets* on page 89.

An Ebix is a collection of metadata that defines the structure of data. The Ebix supplied with the iWay Integration Solution for SWIFT defines the structure of supported SWIFT messages.

Each Ebix includes:

- Pre-built data dictionaries. The structure of each SWIFT document is described by the dictionary, which describes the segments and elements that compose each document.

  The dictionaries from the Ebix are used to transform the structure of a document per SWIFT standards.

- Pre-built XML schemas that define the structure and content of XML messages in detail.

- Pre-built SWIFT to XML transformation templates, and XML to SWIFT templates, for the supported SWIFT FIN messages.

- Pre-built rule files for each SWIFT message. The iWay Integration Solution for SWIFT uses these rule files to validate inbound and outbound documents for network rule compliance and other data checking.

Preparsers

A preparser is an iWay business component that converts incoming messages into processable documents.

Typically a preparser converts a non-XML document into XML format. The preparser for the iWay Integration Solution for SWIFT converts an incoming SWIFT message to XML format.

There are four preparsers that are provided with the iWay Integration Solution for SWIFT:

- XDSWIFTPreParser (com.ibi.preparsers.XDSWIFTPreParser)
  - Recommended preparser to use for inbound SWIFT processing.
  - Processes a single SWIFT message through a SWIFT inbound channel.
  - Transforms system messages to XML, and will create output as any other documents.
  - Use in combination with XDSWIFTValidationReportAgent to produce a SWIFT validation report.
  - Channel configuration requires an Ebix attachment.

The following preparsers can be used in specific situations:

- SWIFTACKPreparser (com.ibi.preparsers.SwiftBPP)
  - Expected inbound SWIFT contains only SWIFT ACK (F21) messages.
- XDSWIFTBatchSplitter (com.ibi.preparsers.XDSWIFTBatchSplitter)
  - Expected inbound SWIFT data contains multiple SWIFT messages.
  - SWIFT inbound channel should include:
    - SWIFTACKPreparser and the SWIFTPreparser.
    - XDSWIFTPreParser, which processes SWIFT ACK messages and standard messages.
- SWIFTSystemMessagePreParser (com.ibi.preparsers.XDSWIFTSysMsgPreParser)
  - Expected inbound SWIFT data contains only SWIFT system messages.

  Note: SWIFT system messages do not require any Ebix. As a result, their transformation cannot be modified.

Preemitter

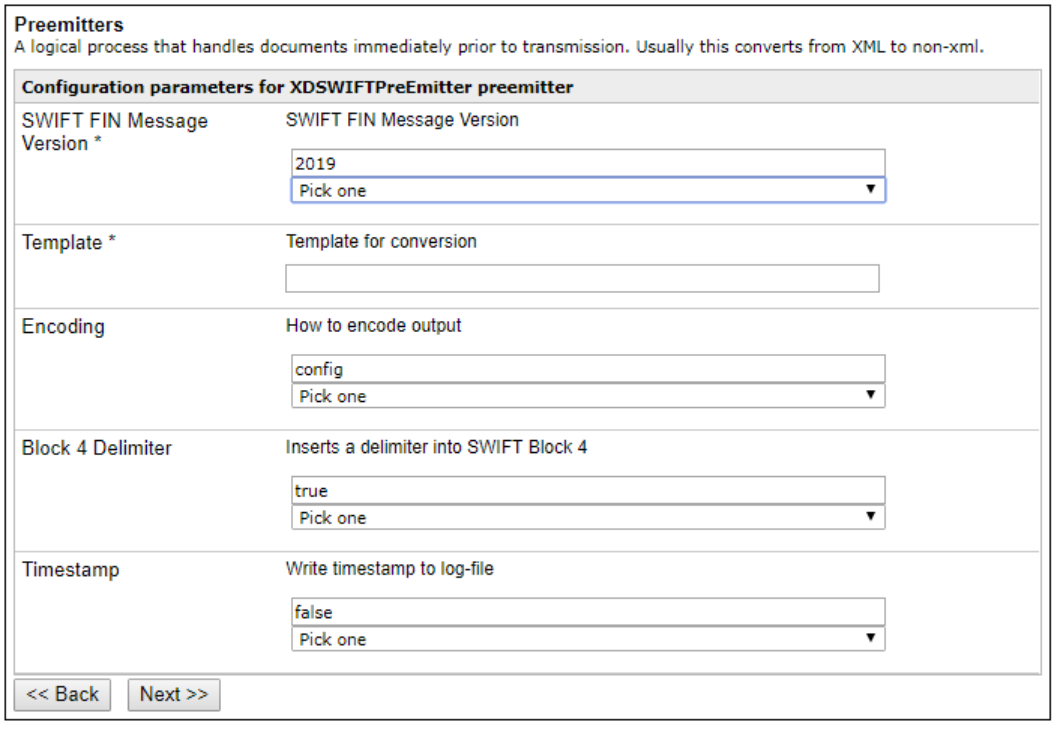
A preemitter is a logical process that handles a document immediately before transmission.

Typically a preemitter is used to convert an XML document to non-XML format. The XML document is created from SWIFT input data in inbound processing. The iWay Integration Solution for SWIFT uses a preemitter in outbound processing to convert the XML-formatted SWIFT document to a SWIFT formatted document.

The XML structure must be compliant with the schema supplied in the Ebix.

The preemitter that is provided for the iWay Integration Solution for SWIFT is the XDSWIFTPreEmitter (com.ibi.preemit.XDSwiftPreEmitter). An Ebix attachment is required for the iSM channel.

The following image shows the available configuration parameters for the SWIFT preemitter in the iSM Administration Console.



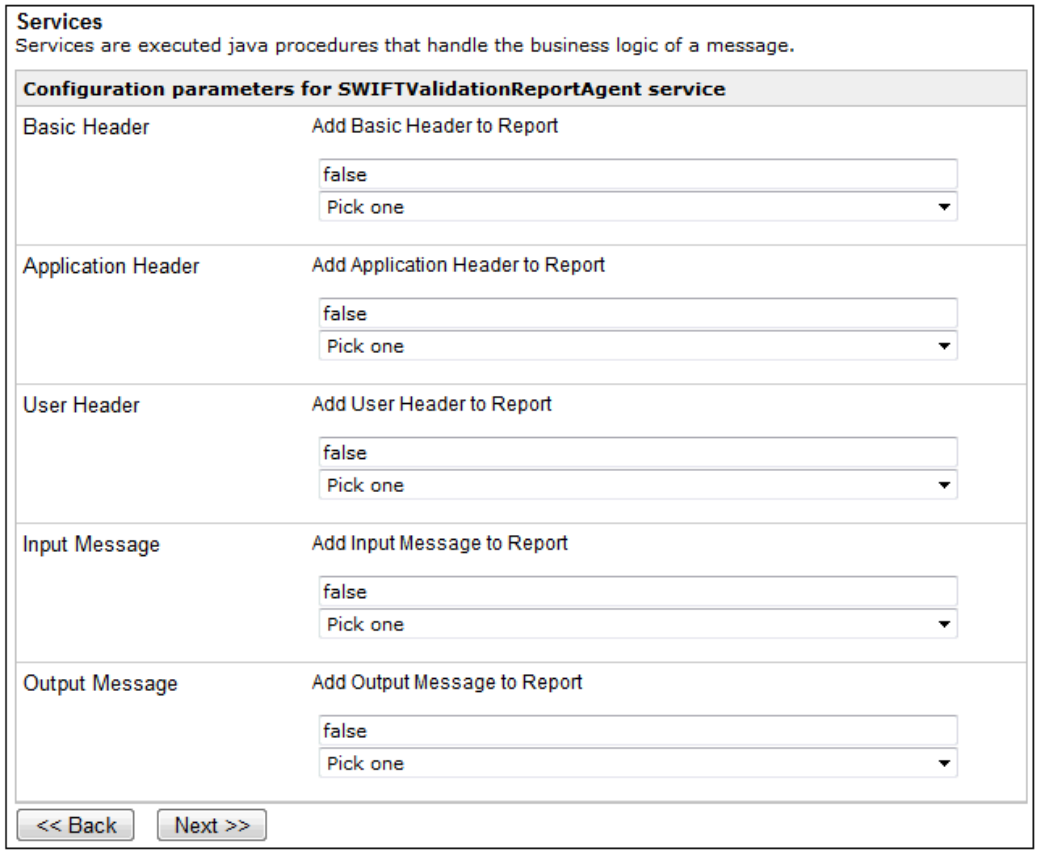
Validation Report Service

There is a validation report service that is provided with the iWay Integration Solution for SWIFT. The validation report is structured in XML format. It contains the input document, the output document, and indicates whether the message is compliant with the SWIFT defined standards.

- XDSWIFTValidationReportAgent (com.ibi.agents.XDSWIFTValidationReportAgent)
  - Used for inbound and outbound SWIFT processing.
  - Reports the results of structural validation from Transformer and Network Validation from Rules processing. Outputs an XML report file with the input document, the output document, and the results of validation.

- An Ebix attachment is required for the iSM channel to invoke validation.

The following image shows the available configuration parameters for the SWIFT validation report service in the iSM Administration Console.



iWay Service Manager Channel Configuration

This section provides sample use case iSM channel configurations.

Use Case	Attach Ebix?	iSM Components
Inbound processing with SWIFT rules validation	Yes	Preparser: XDSWIFTPreParser (com.ibi.preparsers.XDSWIFTPreparser) Service: XDSWIFTValidationReportAgent (com.ibi.agensts.XDSWIFTValidationReportAgent)
Outbound processing with SWIFT rules validation	Yes	Service: XDIWAYSWIFTXXMLTransformAgent (com.ibi.agents.XDIWAYSWIFTXMLTransformAgent) Service: SWIFTValidationReportAgent (com.ibi.agents.XDSWIFTValidationReportAgent)

SWIFT Message Structure

In the SWIFT message structure, blocks 3, 4, and 5 may contain sub-blocks or fields delimited by field tags. Block 3 is optional. Many applications, however, populate this with a reference number so that when the Acknowledgement is returned by SWIFT, it can be used for reconciliation purposes.

Basic Header Block

The basic header block has the following format:

```
{1:    F    01   BANKBEBB   2222   123456}
(a)   (b)  (c)   (d)        (e)    (f)
```

Note: Blank spaces have been added for readability purposes.

The basic header block has a fixed length and is continuous with no field separators:

```
a) Block ID - always '1:'
b) Application ID - F = FIN, A = GPA or L = GPA (logins, etc)
c) Service ID - 01 = FIN/GPA, 21 = ACK/NAK
d) LT address - 12 Characters, must not have 'X' in position 9
e) Session number - added by the CBT (Computer Based Terminal), padded with
zeroes
f) Sequence number - added by the CBT (Computer Based Terminal), padded
with zeroes
```

Application Header Block

The application header block has a different format depending on whether it is being sent to or from SWIFT.

The input structure (to SWIFT) is:

```
{2:    I     100    BANKDEFFXXXX    U       3       003}
(a)    (b)   (c)    (d)             (e)     (f)     (g)
```

Note: Blank spaces have been added for readability purposes.

This structure has a fixed length and is continuous with no field separators from user to SWIFT.

```
a) Block ID - always '2:'
b) I = Input
c) Message Type
d) Receivers address with X in position 9 and padded with X's if no branch
e) Message Priority - S = System, N = Normal, U = Urgent
f) Delivery Monitoring - 1 = Non Delivery Warning (MT010)
                         2 = Delivery Notification (MT011)
                         3 = Both Valid = U1 or U3, N2 or just N
g) Obsolescence Period - when a non delivery notification is generated
   Valid for U = 003 (15 minutes)
   Valid for N = 020 (100 minutes)
```

The output structure (from SWIFT) is:

```
{2:   O      100   1200   970103BANKBEBBAXXX2222123456   970103  1201   N}
(a)  (b)     (c)   (d)    (e)                            (f)     (g)   (h)
```

This structure has a fixed length and is continuous with no field separators from user to SWIFT.

```
a) Block ID - always '2:'
b) O = Output
c) Message Type
d) Input time with respect to the Sender
e) MIR with Senders address
f) Output date with respect to Receiver
g) Message priority
```

User Header Block

The user header block has the following structure:

```
{3:  {113:xxxx}  {108:abcdefgh12345678}     }
(a)  (b)         ( c)
```

Note: Blank spaces have been added for readability purposes.

This is an optional block and is similar in structure to system messages.

```
a) Block ID - always '3:'
b) Optional banking priority code
c) Message User Reference MUR used by applications for reconciliation with
ACK
```

Other tags exist such as 119 which can contain the code ISITC on an MT521 or 523 providing the sender and receiver are registered to do so with SWIFT. This forces additional code word and formatting rules validation of the body of the message as laid down by ISITC (Industry Standardization for Institutional Trade Communication).

113. 4 characters (SWIFT x Character Type) Banking Priority: (HYnn) H- Highly Urgent, U- Urgent, N- Normal, Y- Request Receiving, N- Do Not Request Receiving, nn- May or may not be used.

108. 16 characters (SWIFT x Character Type) MUR (Message User Reference). 1234459898ABC used to reconcile with ACKs.

119. 8 uppercase characters or numbers. Bank User Defined.

Text Block or Body

This block is where the actual MTnnn message is specified and is what most users will see. Generally the other blocks are stripped off before presentation. The format is as follows:

```
{4:
:20:123456
:25:123-456789
:28C:102
:60F:C020527EUR3723495,
:61:020528D1,2FBNK494935/DEV//67914
:61:020528D30,2NCHK78911//123464
:61:020528D250,NCHK67822//123460
:61:020528D450,S103494933/DEV//P064118
FAVOUR K. DESMID
:61:020528D500,NCH45633//123456
:61:020528D1058,47S103494931//3841188
FAVOUR H. F. JASSEN
:61:020528D2500,NCHK56728//123457
:61:020528D3840,S103494935//3841189
FAVOUR H. F. JANSSEN
:61:020528D5000,S20023/200516//47829
ORDER ROTTERDAM
:62F:C020528EUR3709865,13
-}
```

The example above is an MT950, Statement Of Cash. Fields specified are in accordance to the appropriate volume of the user handbook, there is one or more for each message category.

The format of field tags is:

```
:nna:
nn - numbers
a - optional letter which may be present on selected tags
eg - :20:  Transaction Reference Number
     :25: Account Identification
The length of a field is designated thus:
nn      -   maximum length
nn!     -   fixed length
nn-nn   -   minimum and maximum length
nn * nn -   max number of lines times max
            line length
```

The format of the data is designated thus:

```
n      -   Digits
 d      -   Digits with decimal comma
 h      -   Uppercase hexadecimal
 a      -   Uppercase letters
 c      -   Uppercase alphanumeric
 e      -   Space
 x      -   S.W.I.F.T. character set
 y      -   Upper case level A ISO 9735 characters
 z      -   S.W.I.F.T. extended character set
```

Some fields are defined as optional and if not required they must not be present as no blank fields must be present in the message.

```
/,word  -   Characters as-is
 […]        -   optional element
```

For example:

```
4!c[/30x]   -   fixed 4 uppercase alphanumeric, optionally followed by a
slash
                and up to 30 S.W.I.F.T. characters
ISIN1!e12!c -   code word followed by a space and fixed 12 uppercase
alphanumerics
```

In some message types certain fields will be defined as conditional, for example, if a certain field is present then another field may change from optional to mandatory or forbidden.

Certain fields have different formats dependant on the option which is chosen, which is designated by a letter after the tag number, for example:

```
:32A:000718GBP1000000,00    Value Date, ISO Currency and Amount
:32B:GBP1000000,00          ISO Currency and Amount
```

It is important to note that the S.W.I.F.T. standards for Amount formats are, no thousand separators and a comma for a decimal separator.

```
:58A:NWBKGB2L                Beneficiary S.W.I.F.T. address
:58D:NatWest Bank            Beneficiary full name and address
Head Office
London
```

115–> 32 characters (SWIFT x Character Type)SWIFT x Character Types= 0-9,a-z, A-Z, /-?:().,'+

Trailer Block

A message always ends in a trailer with the following format:

```
{5: {MAC:12345678}{CHK:123456789ABC}
```

It contains a number of fields that are denoted by keywords such as:

- MAC: Message Authentication Code calculated based on the entire contents of the message using a key that has been exchanged with the destination and a secret algorithm. Found on message categories 1,2,4,5,7,8, Most 6's and the 304.
- CHK: Checksum calculated for all message types.
- PDE: Possible Duplicate Emission added if user thinks that they may have sent the message previously.
- PDM: Possible Duplicate Message added by S.W.I.F.T. if they think that a message may have been transmitted previously.
- DLM: Added by S.W.I.F.T. if an Urgent message has not been delivered within 15 minutes or a Normal message within 100 minutes.

Validation Report

The Validation Report service contains options that will allow the user to include or suppress the Basic, Application, and User Headers from the report.

Chapter 3

Configuring the EDI Activity Driver

This section describes how to configure the EDI Activity Driver using iWay Service Manager.

In this chapter:

EDI Activity Driver Overview

The EDI Activity Driver is an extension of the Activity Facility in iWay Service Manager. It is used to log events as messages are processed. Logging can occur when:

- Messages are acquired.
- Messages are emitted.
- Errors occur.
- Components such as a service (agent) or process flow are called.

For more information about the Activity Facility, see the *iWay Service Manager User's Guide*.

Using iWay Service Manager, you must first configure the EDI data provider and then the Activity Facility handler.

Configuring the EDI Data Provider Using iWay Service Manager

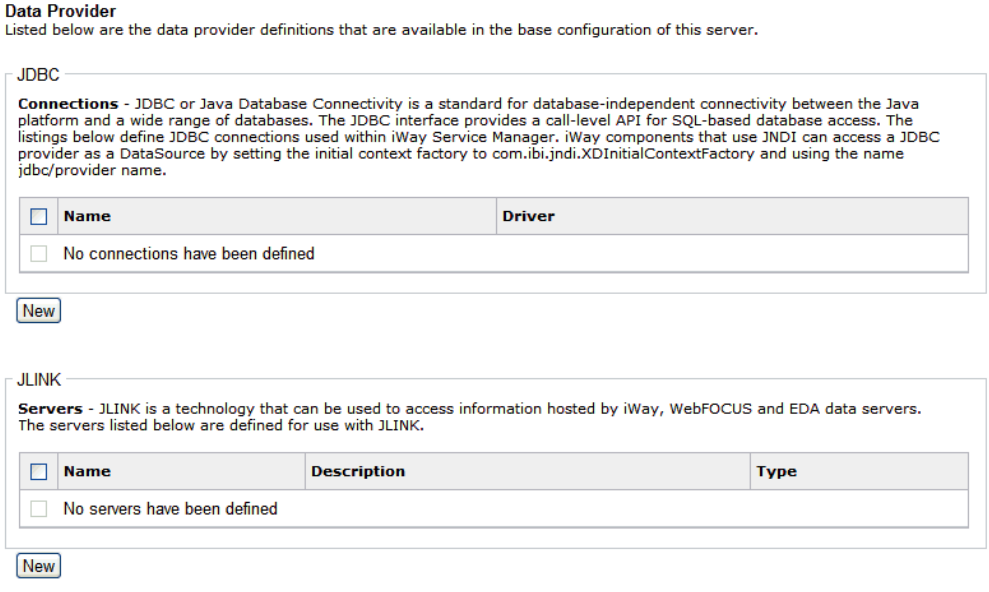
This section describes how to configure the EDI data provider.

Procedure: How to Configure the EDI Data Provider

To configure the EDI data provider:

1. In the left console pane of the Server menu, select *Data Provider*.

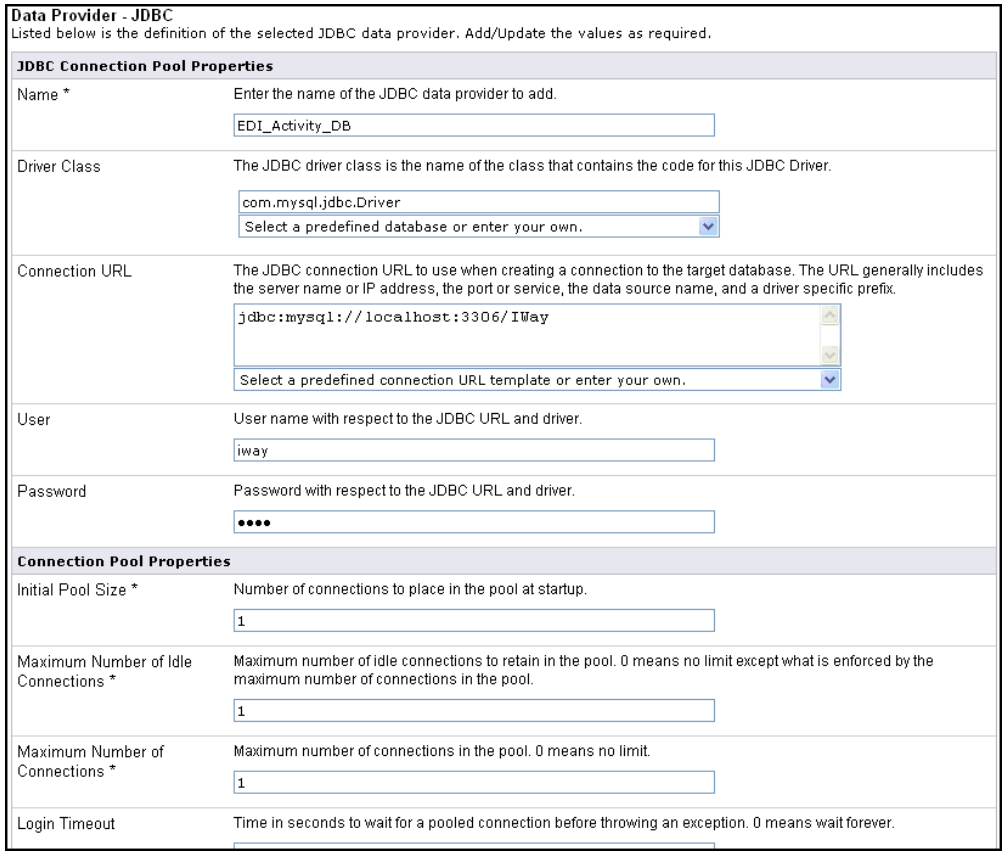
   The Data Provider pane opens.



   The tables that are provided list the configured JDBC and JLINK data providers that are available. By default, no data providers are configured.

2. In the JDBC area, click *New* to configure a new JDBC data provider.

The configuration pane for the JDBC data provider opens.



3. In the Name field, enter a name for the new JDBC data provider, for example, EDI_Activity_DB.
4. From the Driver Class drop-down list, select an appropriate driver or enter the specific driver name (class) that you are using, for example:

   ```
   com.mysql.jdbc.Driver
   ```

5. From the Connection URL drop-down list, select an appropriate connection URL or enter the specific driver connection URL that you are using, for example:

   ```
   jdbc:mysql://localhost:3306/IWay
   ```

6. In the User field, enter a user name with respect to the JDBC URL and driver.
7. In the Password field, enter a password with respect to the JDBC URL and driver.
8. In the Initial Pool Size field, enter the number of connections to place in the connection pool during startup.

9. In the Maximum Number of Idle Connections field, enter the maximum number of idle connections to retain in the connection pool.

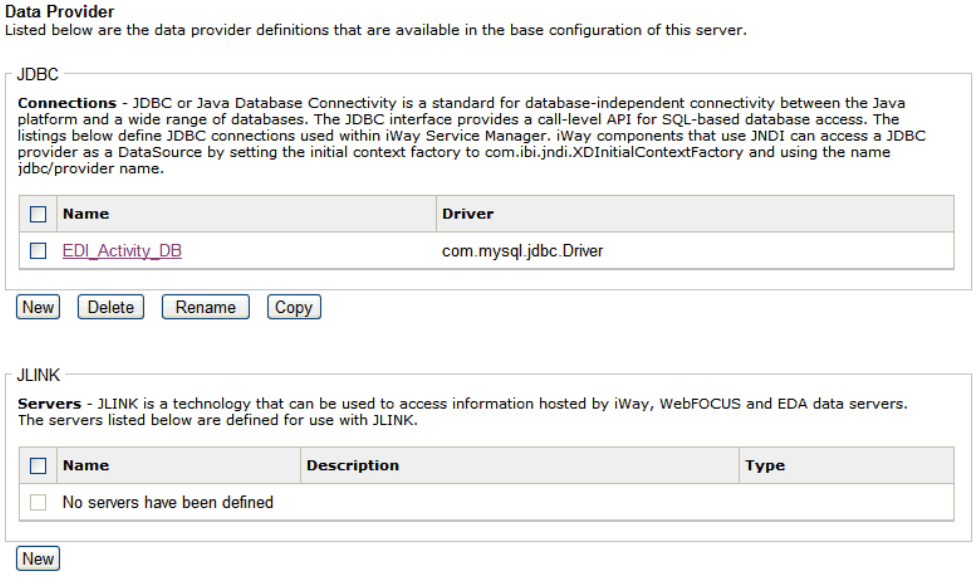
   A value of zero means that there is no limit, except what is enforced by the maximum number of connections in the connection pool.

10. In the Maximum Number of Connections field, enter the maximum number of connections in the connection pool.

    A value of zero means that there is no limit.

11. Click *Add*.

    The JDBC data provider that you configured is added to the JDBC Connections list, as shown in the following image.

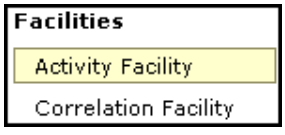


Configuring the EDI Activity Driver Using iWay Service Manager

This section describes how to configure the EDI Activity Driver.

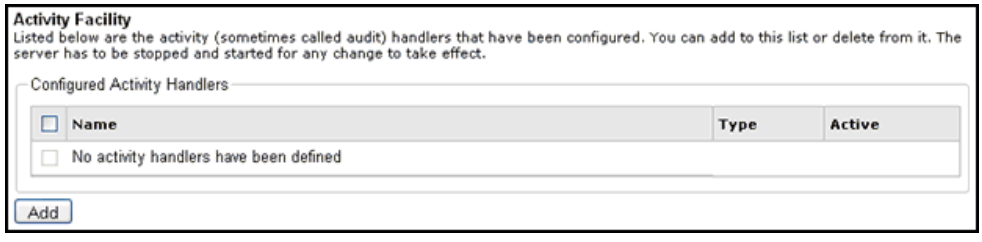
Procedure: How to Configure the EDI Activity Driver

To configure the EDI Activity Driver:



1. In the left console pane of the Server menu, select *Activity Facility*.

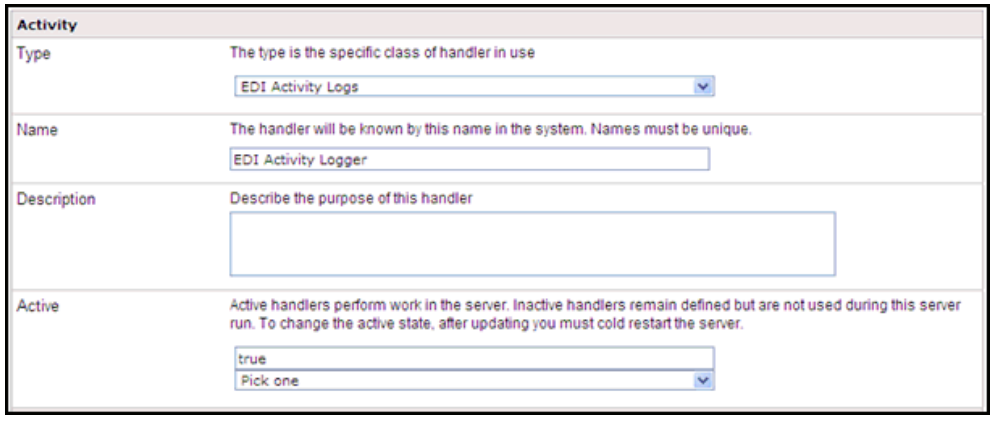
   The Activity Facility pane opens.



   The table that is provided lists the configured Activity Facility handlers. Initially, no handlers are shown.

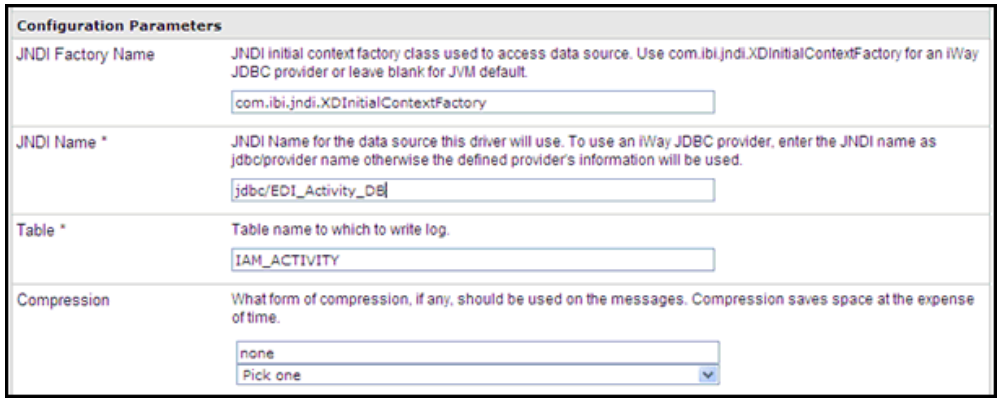
2. Click *Add* to configure a new Activity Facility handler.

   The configuration pane for the Activity Facility handler opens.



3. From the Type drop-down list, select *EDI Activity Logs*.
4. Enter a unique name for the EDI Activity Driver and a brief description.
5. From the Active drop-down list, select *true*.

6. Configure the JDBC driver for the database you are using.



If the database tables do not exist, they will be automatically created when the iSM is restarted.

7. Provide values for the remaining parameters, as defined in the following table.

Parameter Name	Type	Description
JNDI Factory Name	String	The JNDI initial context factory class that is used to access a data source. Use `com.ibi.jndi.XDInitialContextFactory` for an iWay JDBC provider or leave this field blank for the JVM default.
JNDI Name	String	The JNDI name for the data source this driver will use. To use an iWay JDBC provider, enter the JNDI name as `jdbc/<data provider name>`, where *`data provider name`* is the name of the EDI Activity Driver that was specified in step 4. Otherwise the information for the defined provider will be used.
Table	String	Table name for the activity log. This must be a valid identifier in the database being used. If the table does not exist at startup, it will be created automatically.

Parameter Name	Type	Description
Compression	Drop-down list	Specify whether the messages are to be compressed. Values include: ❑ none (default) ❑ smallest ❑ fastest ❑ standard ❑ Huffman
Start Events	Boolean Drop-down list	If set to *true* (default), the input messages will be recorded in the activity log. These values must be set to *true* for use of the audit reports in the console.
Internal Events	Boolean Drop-down list	If set to *true*, system events are included in the activity log. System events include activities such as parsing and transformations (optional). False is selected by default.
Security Events	Boolean Drop-down list	If set to *true* (default), security events are recorded. This includes digital signature, and so on. However, console activity is not recorded.
Business Error Events	Boolean Drop-down list	If set to *true*, business errors are recorded, such as rules system violations. False is selected by default.
Emit Events	Boolean Drop-down list	If set to *true* (default), output messages from emitter services will be recorded. This is required for use of the audit log reports in the console.

Parameter Name	Type	Description
End Events	Boolean Drop-down list	If set to *true* (default), the end of message processing will be recorded in the activity log. This is required for use of the audit log reports in the console.
Notes Table	String	Table name for the notes table, which contains log annotations. If the table does not exist at startup, it will be created automatically.
MAC Algorithm	String Drop-down list	The Message Authentication Code (MAC) algorithm. None (default) indicates a MAC should not be computed.
MAC Provider	String Drop-down list	The Message Authentication Code (MAC) provider. Not Specified indicates the default provider should be used. The remaining available value is *SunJCE*.
MAC Secret Key	String	The Message Authentication Code (MAC) secret key to use.

8. Click *Update*.

   If necessary, start the database services.

9. Restart iSM to start the EDI Activity Driver and begin logging.

   The EDI Activity Driver inserts records into the configured activity database. The records are designed for fast writing rather than for ease of later analysis. A set of inquiry service agents suitable for use in a process flow is available to assist during the analysis of the log. Users are cautioned that iWay does not guarantee the layout of the record from release to release, and this should be checked against the actual schema.

Database Field	Description
recordkey	Unique record identifier.

Database Field	Description
recordtype	Type of this record - the event being recorded. ❏ 101 - Message start. ❏ 131 - Entry to event (see subtype codes below). ❏ 132 - Normal exit from event. ❏ 133 - Failed exit from event. ❏ 151 - Ancillary message (usually rules violation). ❏ 181 - Emit. ❏ 191 - Message end.
signature	Encoding of the listener name and protocol.
protocol	Name of the protocol.
address	Address to which an emit is to be issued. The format depends on the protocol.
tstamp	Timestamp of record.
correlid	The Message Control ID assigned to this message.
tid	The Transaction ID assigned to this message.
msg	Message appropriate to this record type. For example, an input message contains the original message received, if possible. Streaming input does not contain a record.
context	Serialized special registers that were in the context at the time the record was written.
text	Message text for business errors (rules system violations).

Database Field	Description
status	Status code recorded. ❑ 0 - Success ❑ 1 - Success, message end (191 record) ❑ 10 - Rules error
subtype	Event code for event records. ❑ 1 - Preparser ❑ 2 - Parser ❑ 3 - In reviewer ❑ 5 - In validation ❑ 6 - In transform ❑ 7 - Agent or flow ❑ 8 - Out transform ❑ 9 - Out validation ❑ 11 - Preemitter ❑ 1000 - input record written to table before transformation
partner_to	The name of the receiving partner.
partner_from	The name of the sending partner.
encoding	Encoding of the listener that obtained the document.
mac	Not used in this version.
Driver version	1.0 in iSM.

Chapter 4

Working With SWIFT Inbound and Outbound Sample Applications Using iWay Integration Tools

This chapter describes how to work with SWIFT inbound and outbound sample applications using iWay Integration Tools (iIT).

In this chapter:

Overview

This chapter provides instructions to import and deploy SWIFT inbound and outbound sample applications using iWay Integration Tools (iIT). The sample applications are packaged as an iWay Integration Application (iIA), which you can deploy.

The iIA will be used to transform SWIFT to XML for inbound processing and XML to SWIFT for outbound processing.

The inbound application channel creates an XML representation of a SWIFT inbound message, and an XML-formatted validation report.

The outbound application channel creates a SWIFT message from XML, and a XML-formatted validation report.

Documents are routed to designated folders based on the success or failure results of the transformation and SWIFT validation.

Prerequisites

Before you continue, ensure that the following prerequisites are met:

- ❑ You have a working knowledge of iWay Service Manager (iSM) and iWay Integration Tools (iIT).
- ❑ iSM Version 8.0 or higher is installed.
- ❑ iWay E-Business Adapter for SWIFT is installed.
- ❑ iIT Version 8.0 or higher is installed.

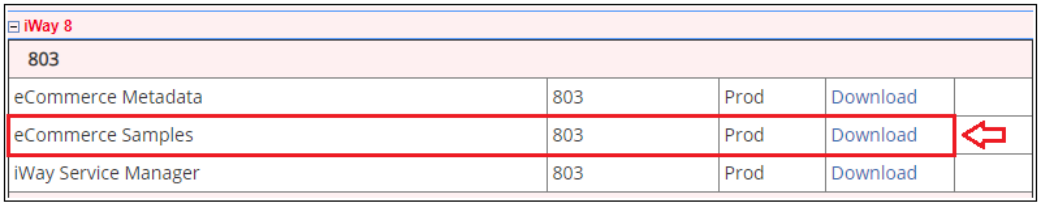
Extracting SWIFT User Directories and Data Samples

This section describes how to extract user data samples for SWIFT.

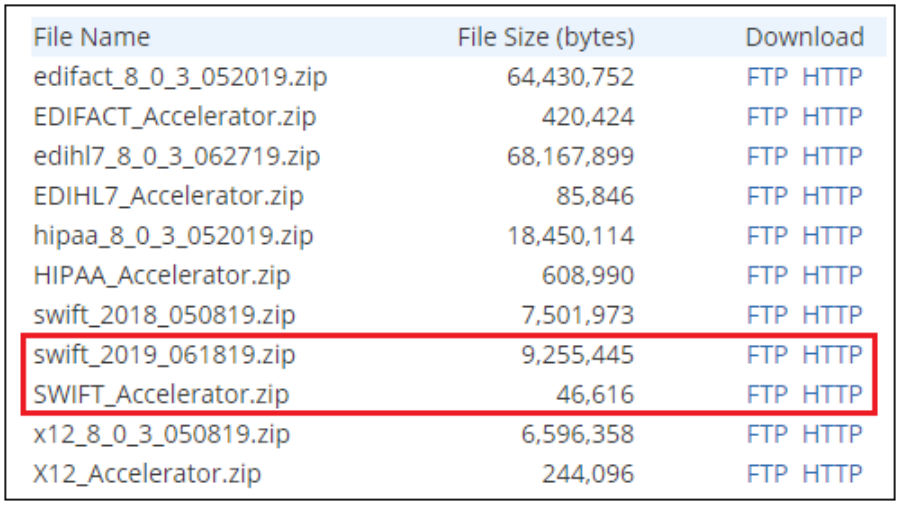
Procedure: How to Extract SWIFT User Directories and Data Samples

1. Download the eCommerce Samples for SWIFT (version 8.0.3) from the Information Builders Technical Support Center:

   https://techsupport.informationbuilders.com/

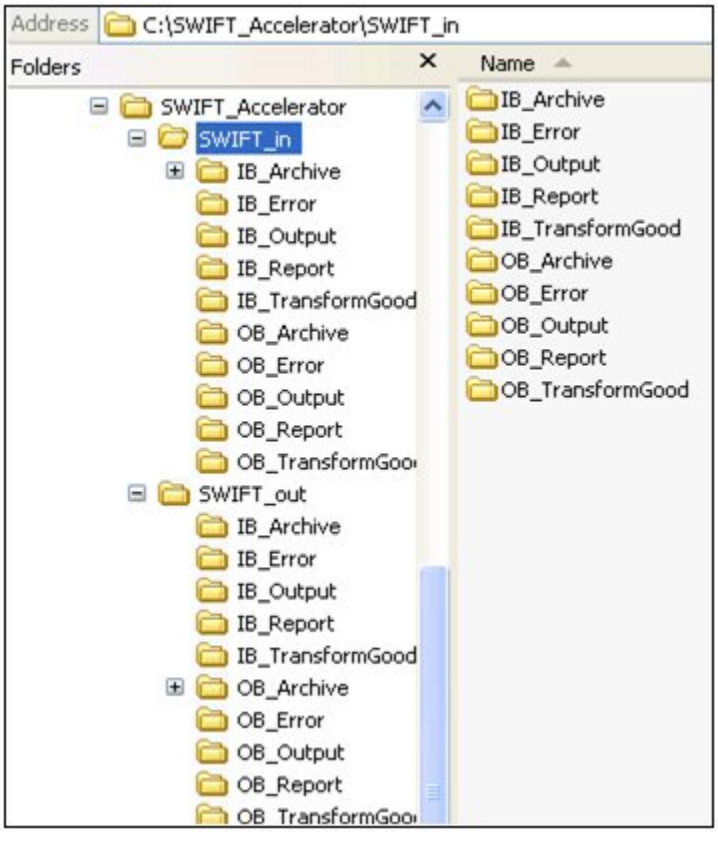


2. Download the *swift_2019_061819.zip* and *SWIFT_Accelerator.zip* files, as shown in the following image.



File Name	File Size (bytes)	Download
edifact_8_0_3_052019.zip	64,430,752	FTP HTTP
EDIFACT_Accelerator.zip	420,424	FTP HTTP
edihl7_8_0_3_062719.zip	68,167,899	FTP HTTP
EDIHL7_Accelerator.zip	85,846	FTP HTTP
hipaa_8_0_3_052019.zip	18,450,114	FTP HTTP
HIPAA_Accelerator.zip	608,990	FTP HTTP
swift_2018_050819.zip	7,501,973	FTP HTTP
swift_2019_061819.zip	9,255,445	FTP HTTP
SWIFT_Accelerator.zip	46,616	FTP HTTP
x12_8_0_3_050819.zip	6,596,358	FTP HTTP
X12_Accelerator.zip	244,096	FTP HTTP

3. Extract the *SWIFT_Accelerator.zip* file to a location where you want to store your inbound and outbound data, as shown in the following image.

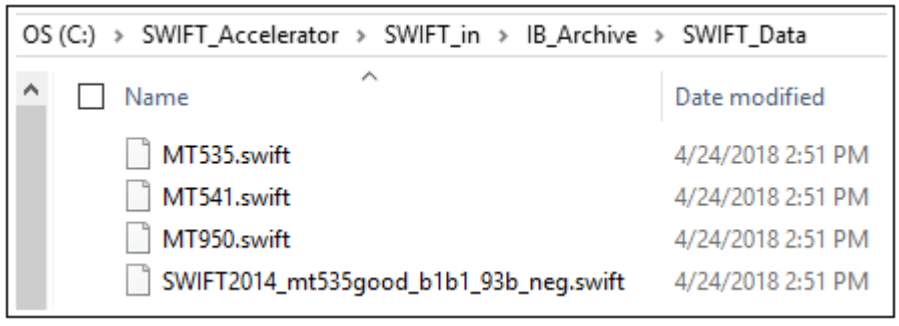


4. The *SWIFT_Accelerator.zip* file contains sample input and output data for SWIFT that you can use.

   ❏ Sample inbound test data (in SWIFT format) is located in the following folder:

   `\SWIFT_Accelerator\SWIFT_in\IB_Archive\SWIFT_Data`

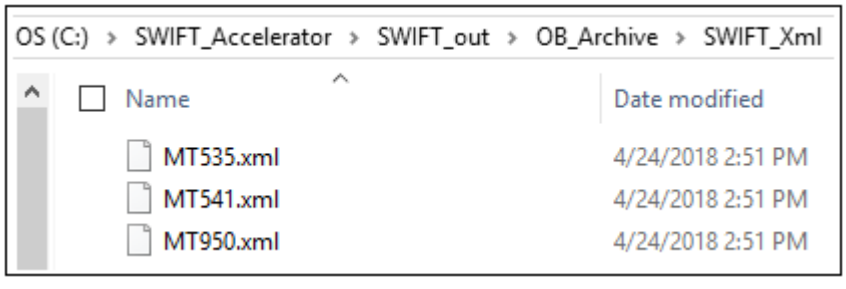
For example:



- Sample outbound test data (in XML format) is located in the following folder:

  `\SWIFT_Accelerator\SWIFT_out\OB_Archive\SWIFT_Xml`

  For example:



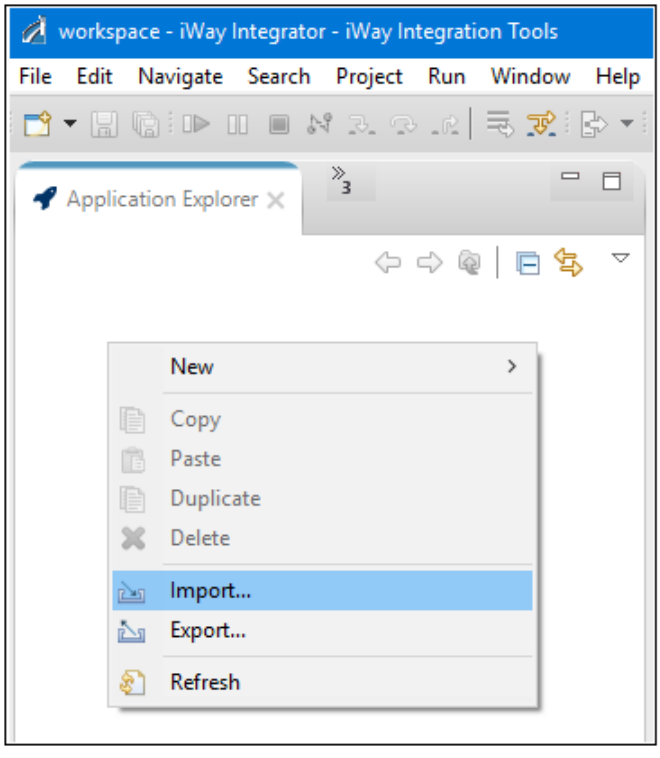
Importing the SWIFT Sample Application Workspace

This section describes how to import the SWIFT sample application workspace into iWay Integration Tools (iIT).

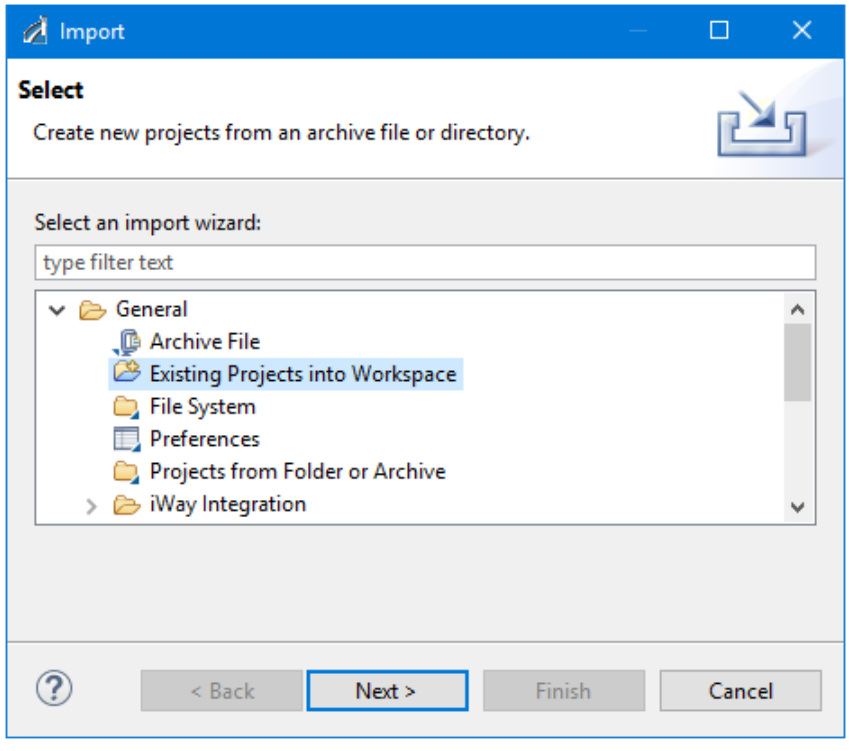
Procedure: How to Import the SWIFT Sample Application Workspace

1. Start iWay Integration Tools (iIT).

2. Right-click anywhere inside the Application Explorer tab and select *Import...* from the context menu, as shown in the following image.

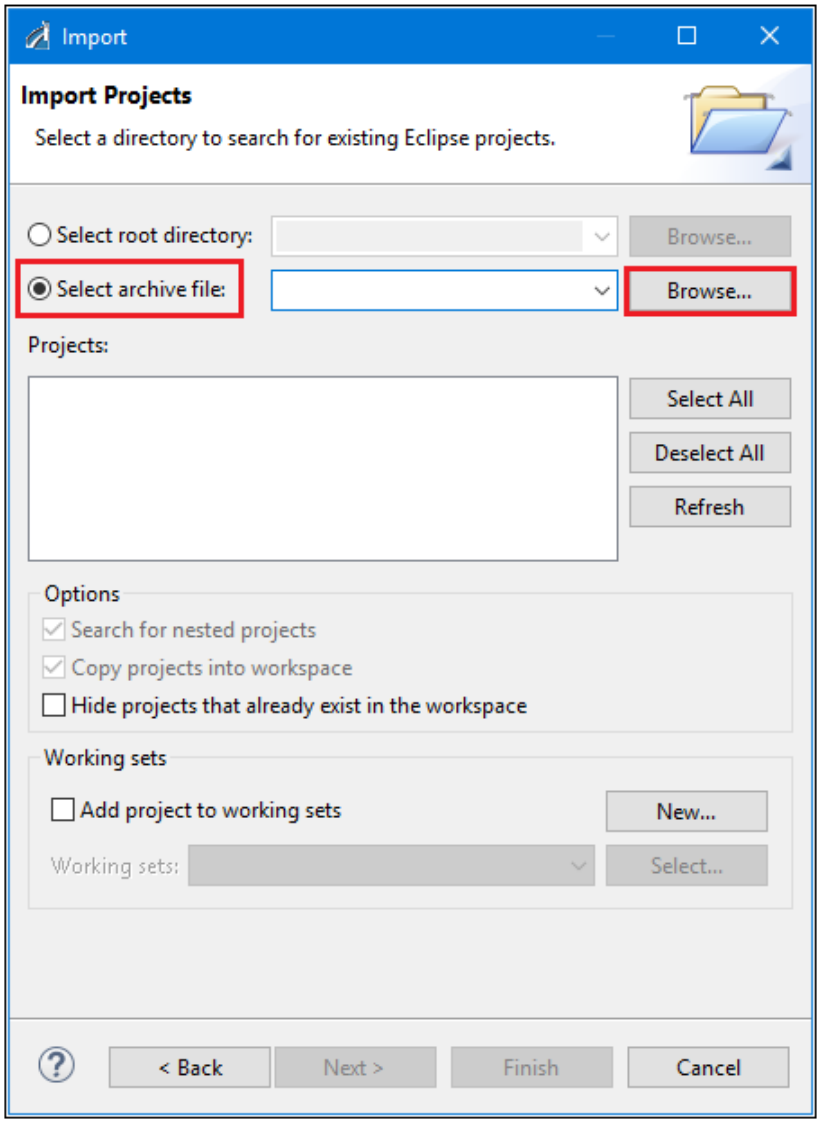


The Import dialog opens, as shown in the following image.



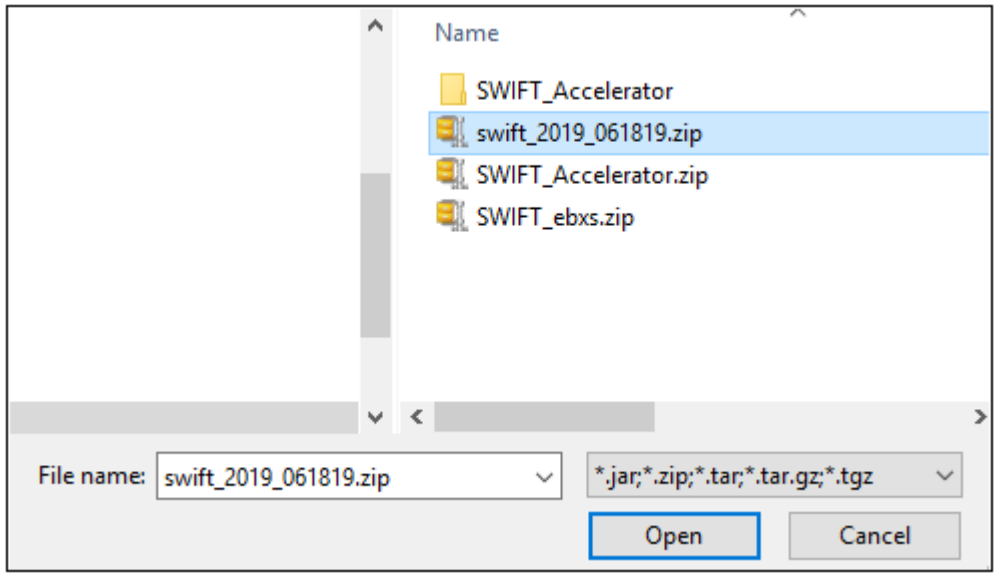
3. Expand the *General* folder, select *Existing Projects into Workspace*, and then click *Next*.

The Import Projects pane opens, as shown in the following image.



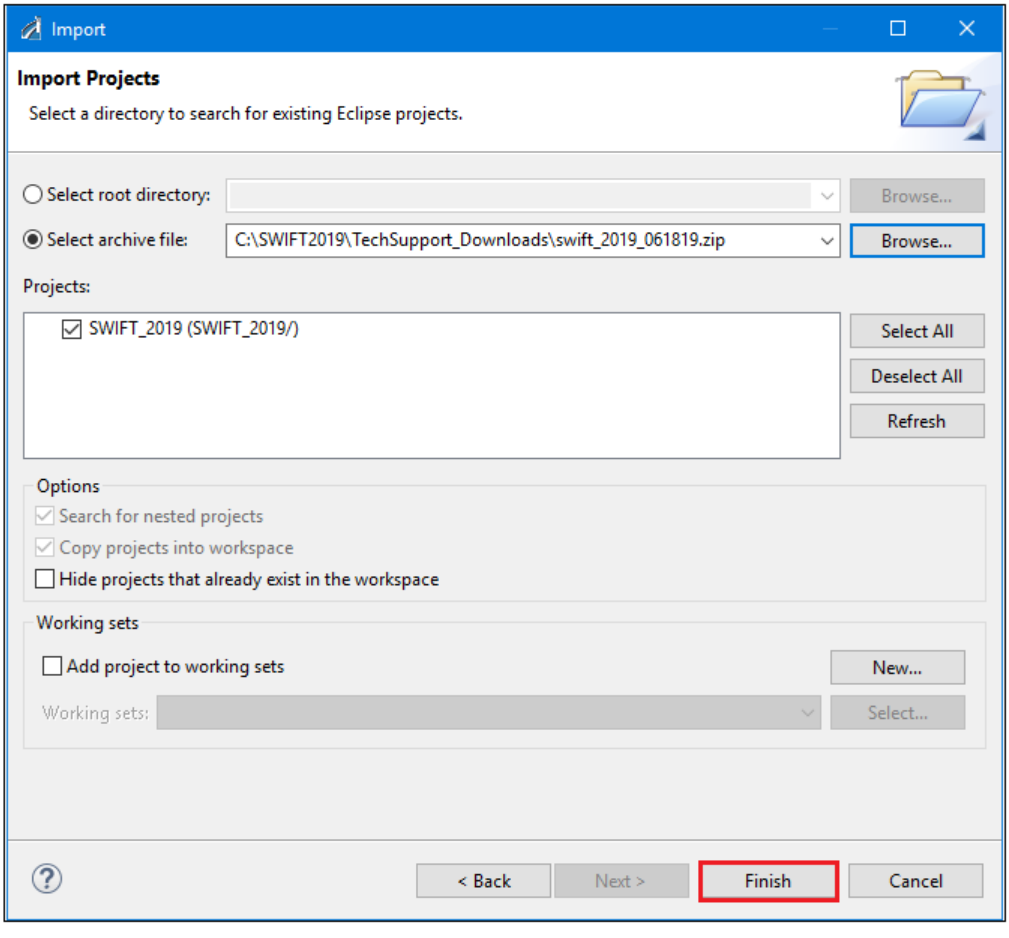
4. Click *Select archive file*, and then click *Browse*.

The Select archive import pane opens.



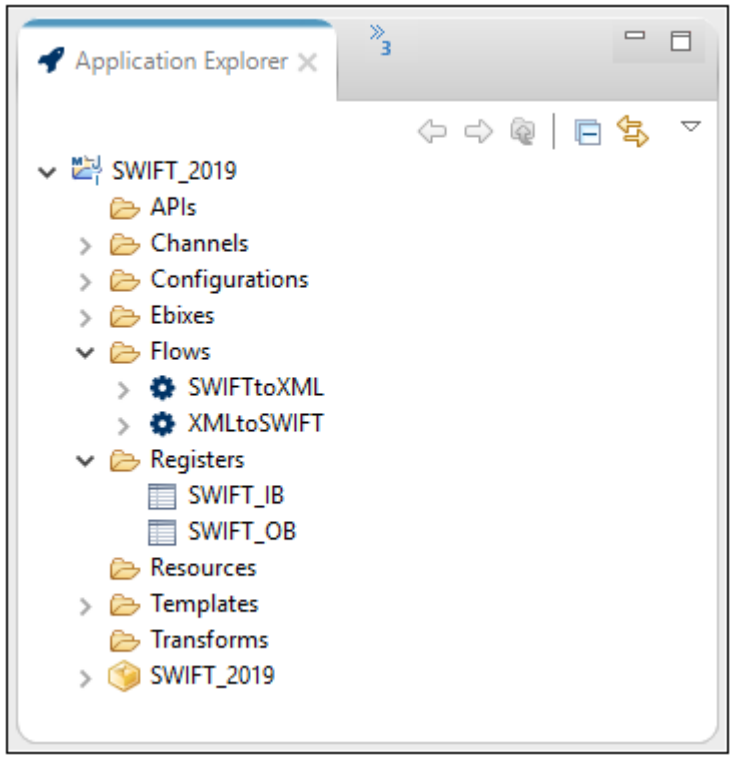
5. Navigate to where you saved the SWIFT 2019 archive file for the sample application (for example, *swift_2019_061819.zip*), select the file and then click *Open*.

You are returned to the Import Projects pane, as shown in the following image.



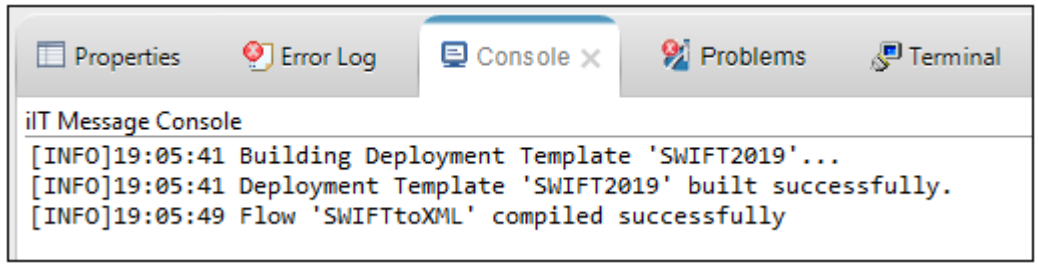
6. Click *Finish*.

The SWIFT 2019 sample application and user samples are loaded into your iIT workspace, as shown in the following image.

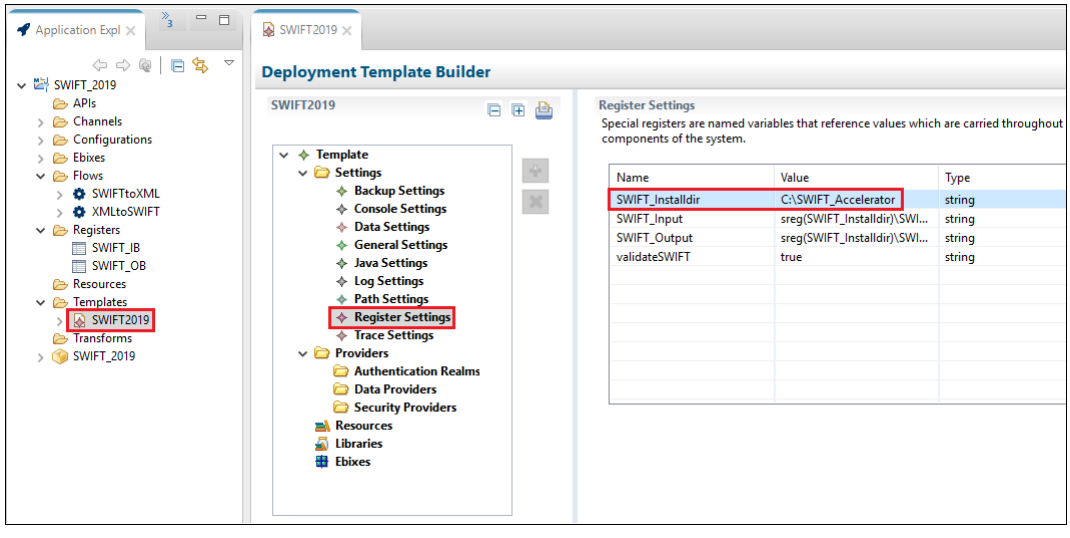


The Application Explorer tab on the left pane displays a hierarchy of all the imported application components (for example, channels, ebixes, process flows, registers, templates, and so on). The Console tab on the bottom provides a status as each channel component is imported.

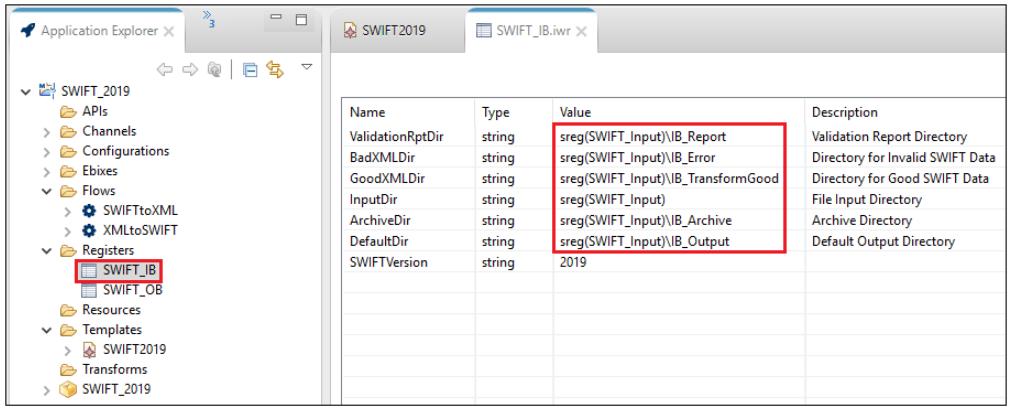
The following messages are displayed in the Console tab, indicating that the import process completed successfully.



You are not required to modify any directory settings, since all of the components that read/write to the directories use the structure that is defined by the *SWIFT_Accelerator.zip* file. Review the SWIFT 2019 template, where you can see how the directories are referenced, as shown in the following image.



The SWIFT 2019 template also references specific registers for SWIFT, as shown in the following image.

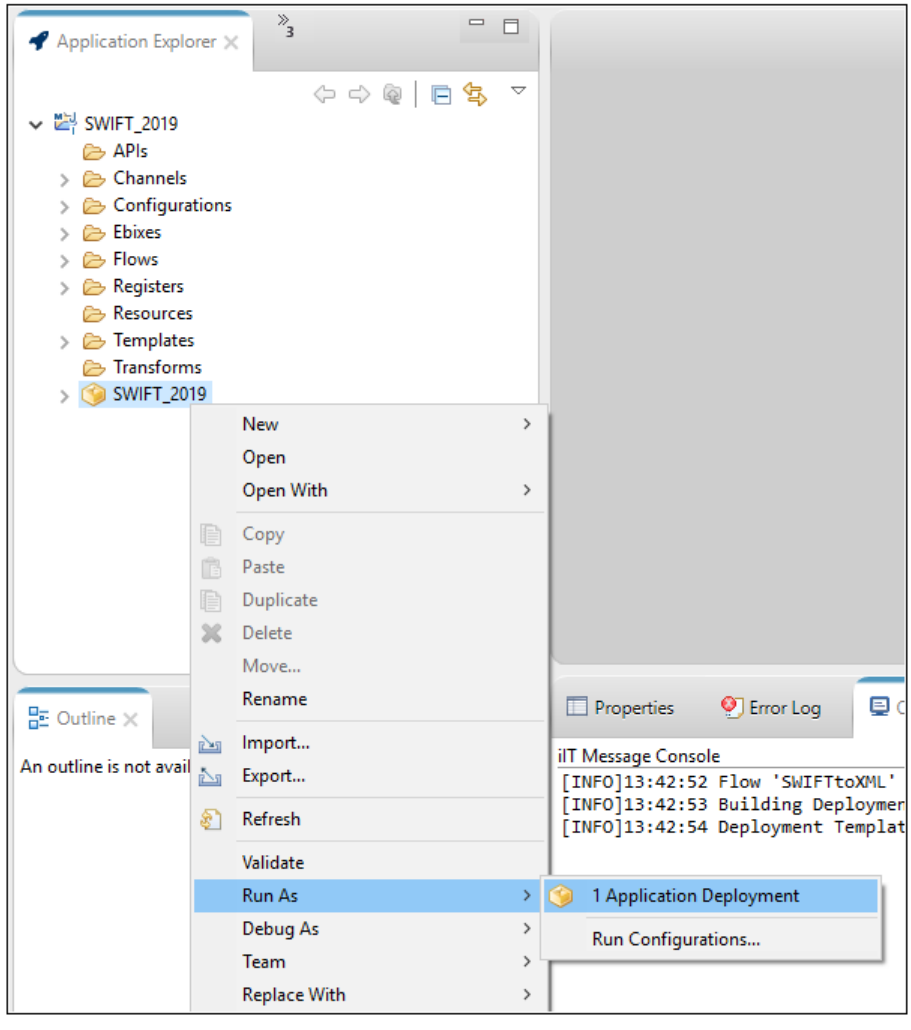


Deploying the iWay Integration Application for SWIFT

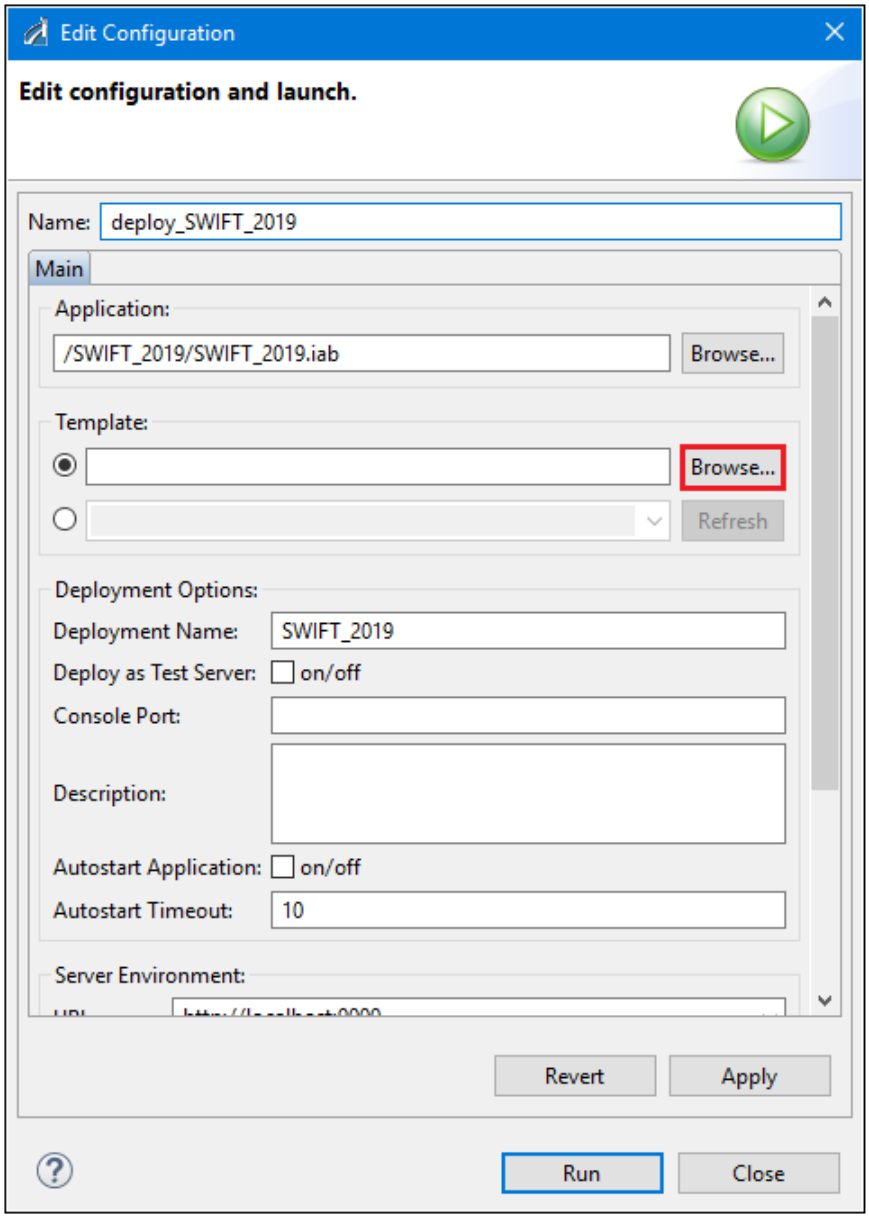
This section describes how to deploy the iWay Integration Application (iIA) for SWIFT.

Procedure: How to Deploy the iWay Integration Application

1. In the Application Explorer tab, right-click *SWIFT_2019*, select *Run As* from the context menu, and then click *1 Application Deployment*, as shown in the following image.

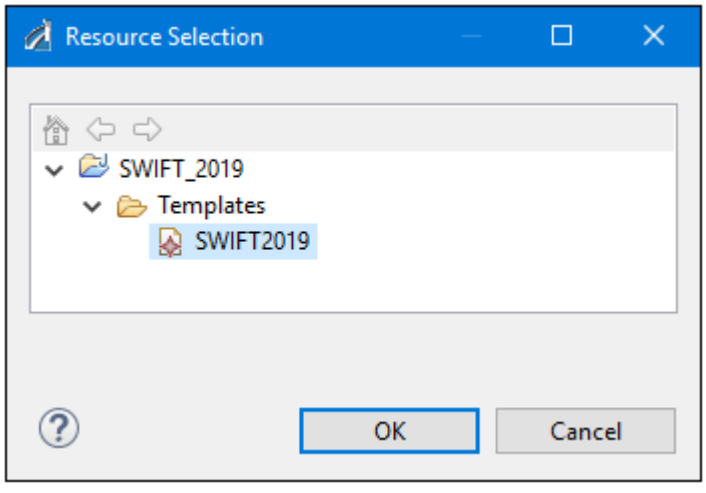


The Edit Configuration dialog opens, as shown in the following image.



2. In the Template field, click *Browse*.

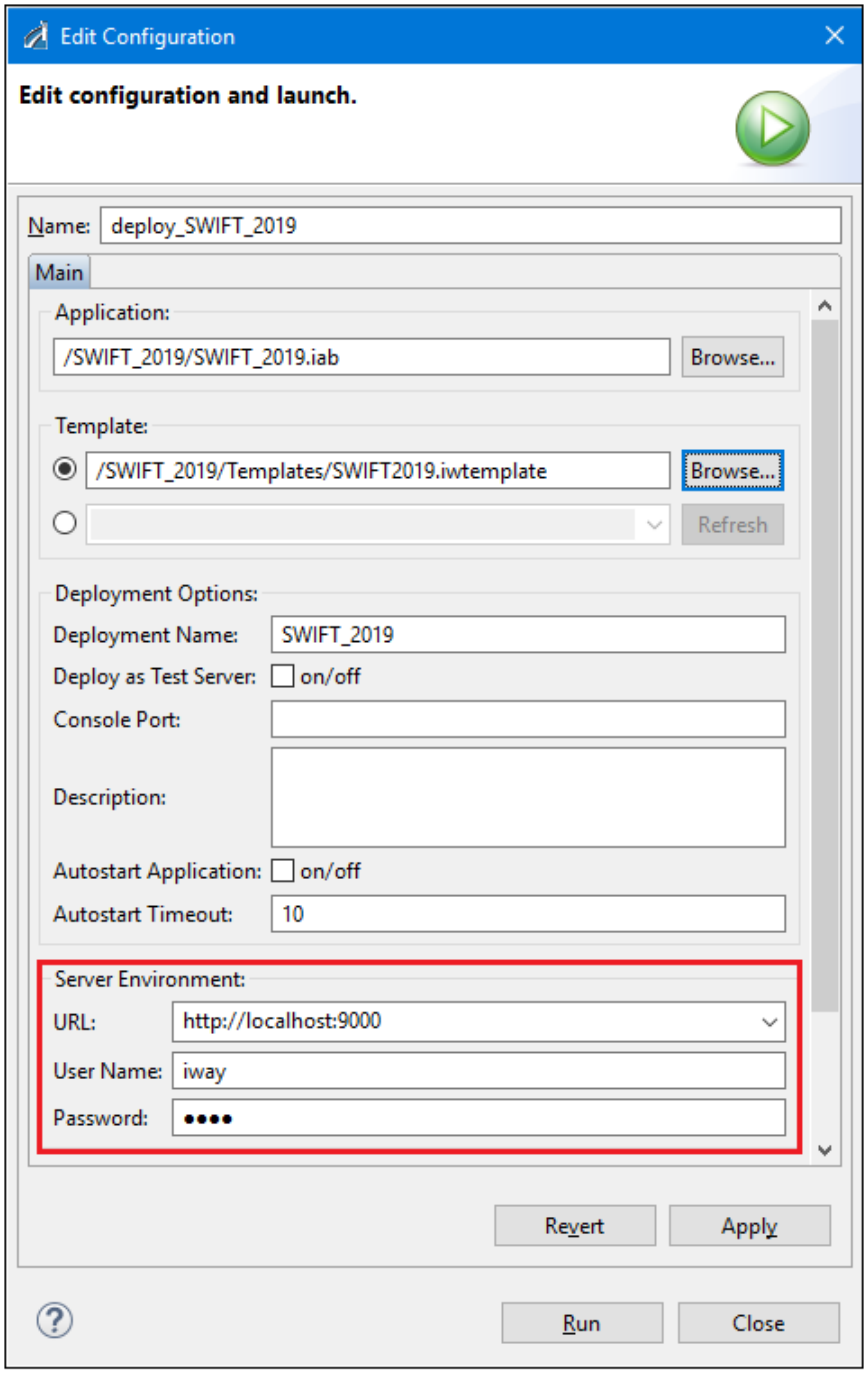
The Resource Selection dialog opens, as shown in the following image.



3. Select the *SWIFT2019* template and then click *OK*.

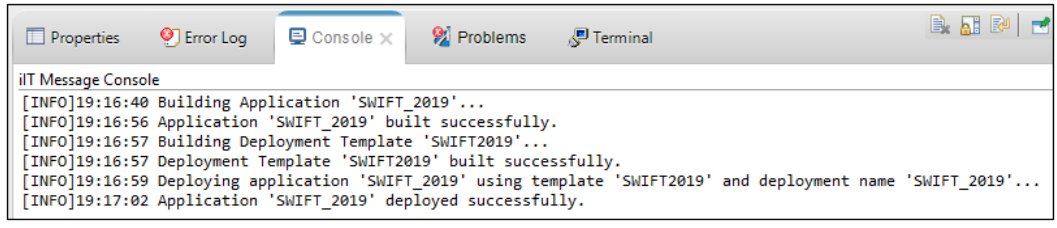
   You are returned to the Edit Configuration dialog.

4. Enter the user name and password for the server, as shown in the following image.



5. Modify the server port if necessary.
6. Click *Run*.

   The following messages are displayed in the Console tab, indicating that the SWIFT application was successfully deployed.

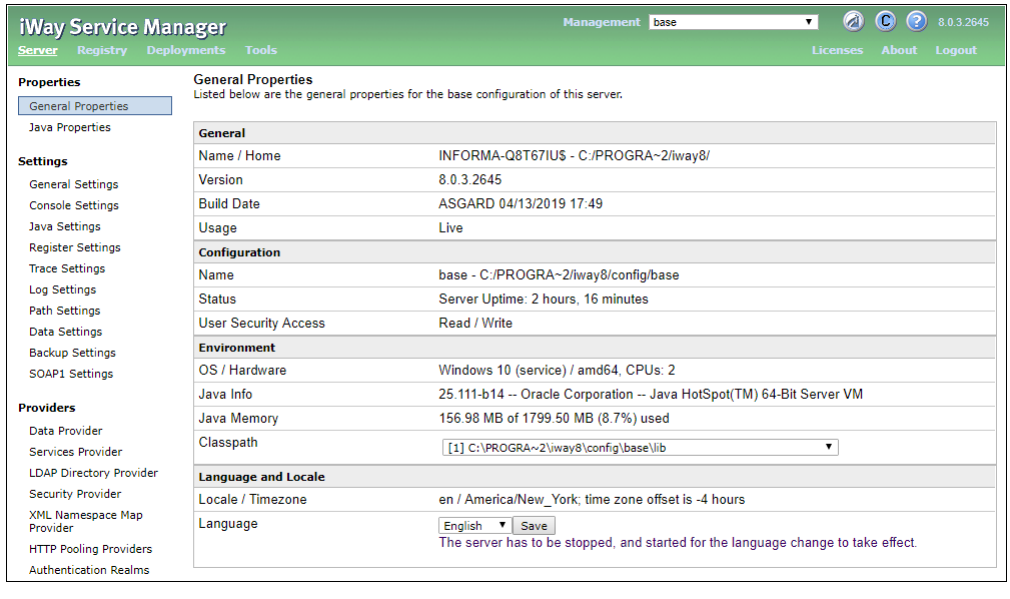


Starting iWay Integration Applications in iWay Service Manager

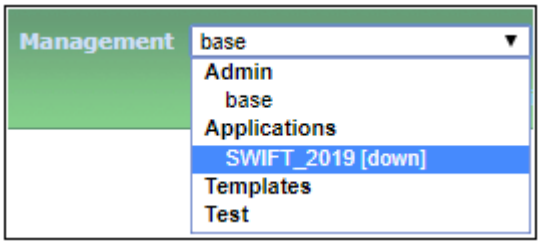
This section describes how to start iWay Integration Applications (iIAs) in iWay Service Manager (iSM).

Procedure: How to Start iWay Integration Applications in iWay Service Manager

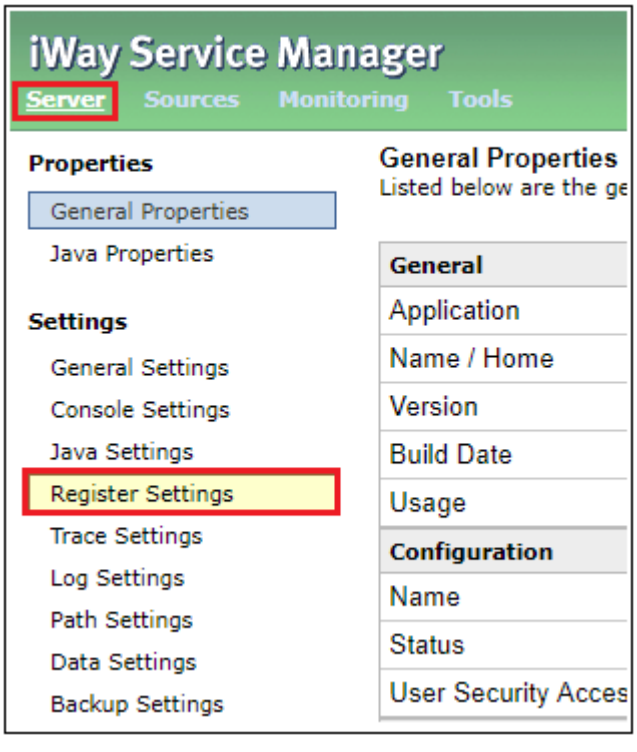
1. Access the iSM Administration Console and sign in using your credentials (user name and password).



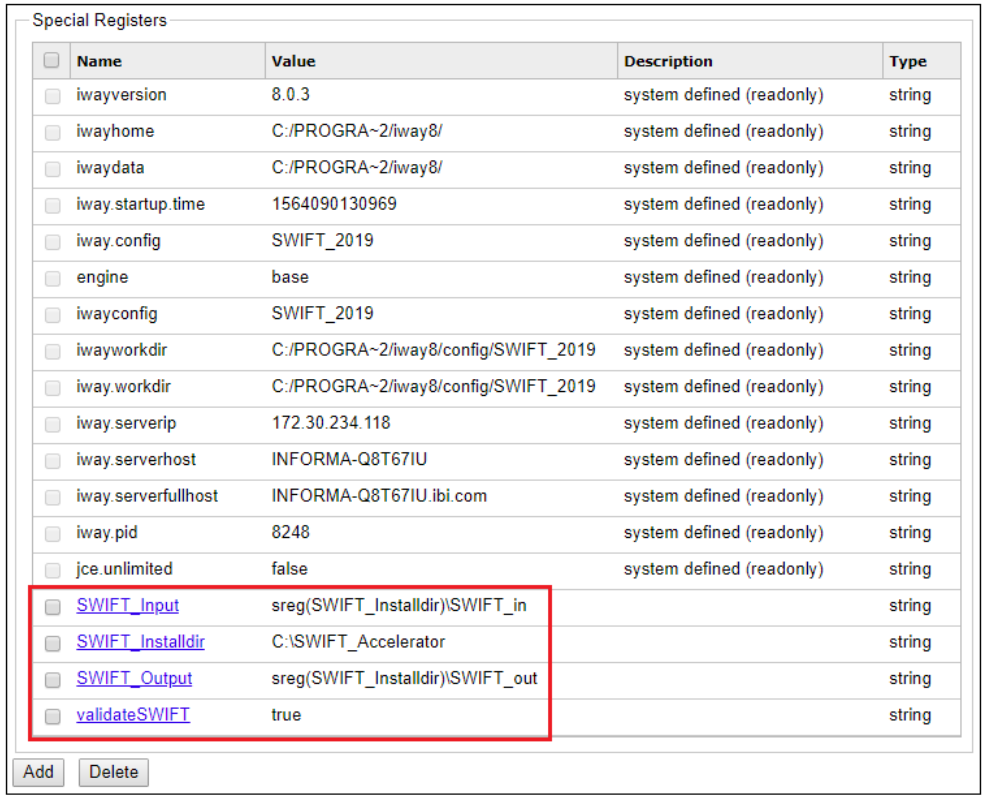
2. Select *SWIFT_2019 [down]* from the Management drop-down list (under Applications), as shown in the following image.



3. Click *Server* from the toolbar and then click *Register Settings* from the left pane, as shown in the following image.

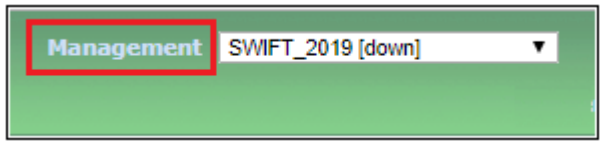


A list of pre-configured system level Special Registers (SREGs) for the SWIFT 2019 application are displayed, as shown in the following image.

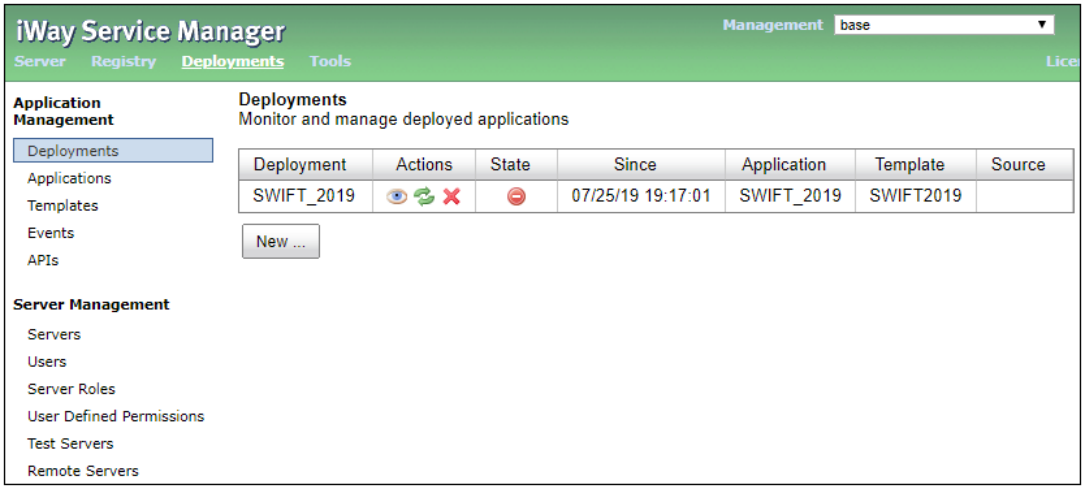


If you did not install the data directories in this location, then click on the SREG and set your location for *SWIFT_Installdir*.

4. Click *Management*, as shown in the following image.

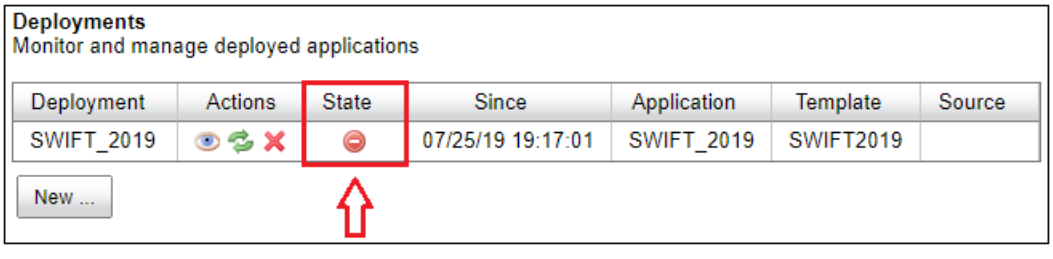


The Deployments pane opens where the deployed SWIFT_2019 application is listed, as shown in the following image.



The SWIFT_2019 application is down (not started), as indicted by the icon in the State column.

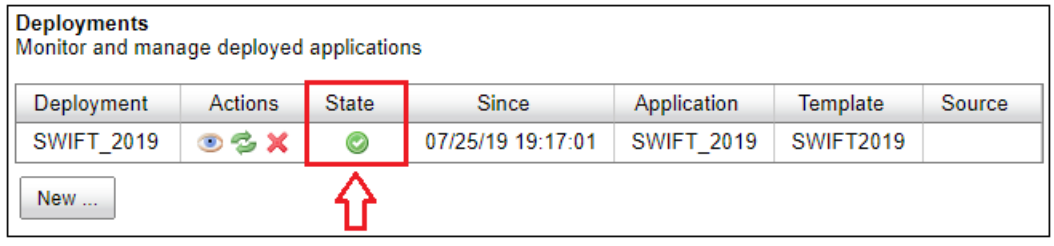
5. In the State column, click the deployment state icon to start the SWIFT_2019 application, as shown in the following image.



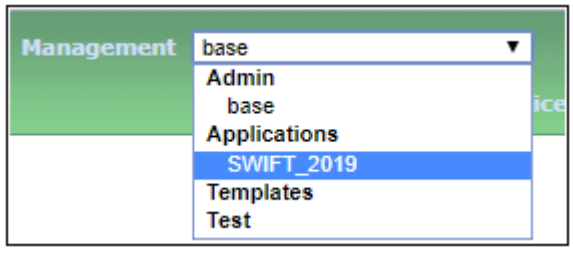
A confirmation message is displayed, as shown in the following image.

6. Click *OK*.

Once the SWIFT_2019 application has successfully started, the icon in the State column changes to a check mark, as shown in the following image.

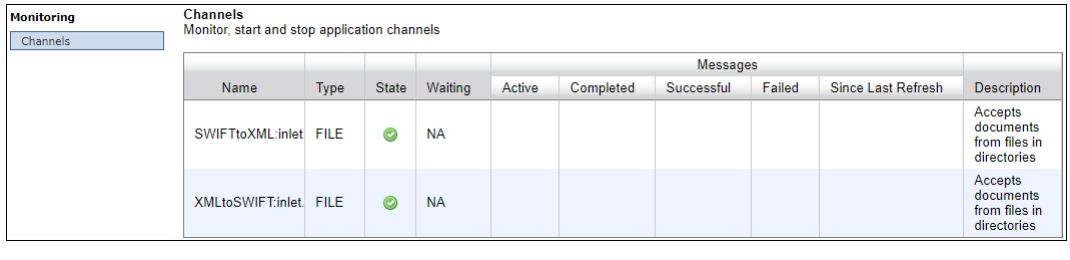


7. Place your input data into the input location that is configured for the SWIFT_2019 application.
8. Select *SWIFT_2019* from the Management drop-down list (under Applications), as shown in the following image.



9. Click *Monitoring* from the toolbar, as shown in the following image.

The deployed application channels *SWIFTtoXML* and *XMLtoSWIFT* are displayed, as shown in the following image.



You can stop or start either channel as required by clicking the green or red icon in the State column.

Testing the Sample Applications

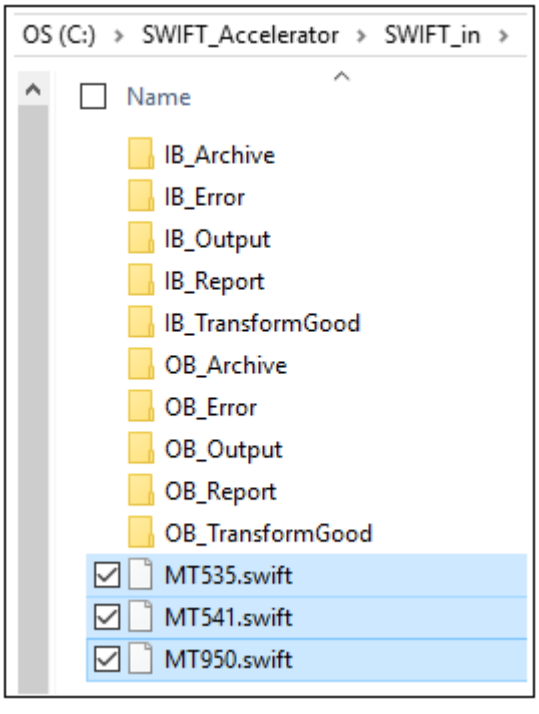
This section describes how to test the sample inbound (SWIFT to XML) and outbound (XML to SWIFT) applications.

Procedure: How to Test the Sample Inbound (SWIFT to XML) Application

1. Copy the sample SWIFT input test data to the following directory:

   `SWIFT_Accelerator\SWIFT_in`

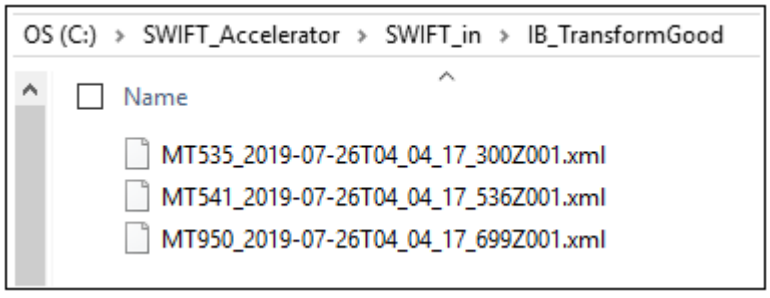
   For example:



2. Observe the transformed XML output in the following directory:

   `SWIFT_Accelerator\SWIFT_in\IB_TransformGood`

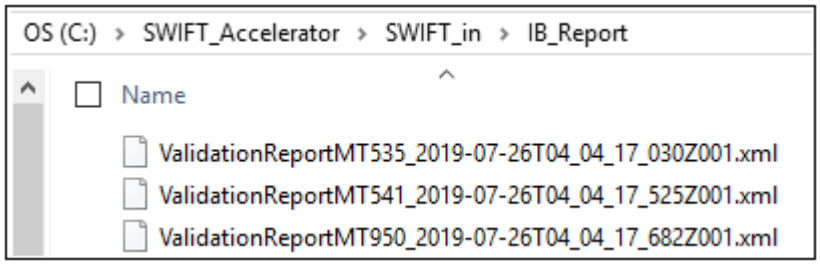
For example:



3. Observe the reports in the following directory:

   `SWIFT_Accelerator\SWIFT_in\IB_Report`

   For example:



4. Observe the acknowledgement in the following directory:

   `SWIFT_Accelerator\SWIFT_in\OB_Output`

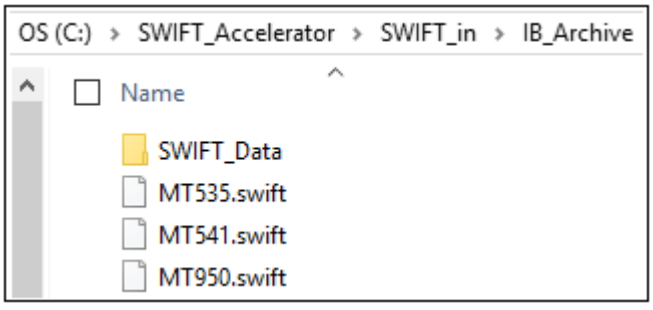
5. If any error occurs during the inbound transformation process, then you can observe error data in the following directory:

   `SWIFT_Accelerator\SWIFT_in\IB_Error`

6. After processing the input data that you place for transformation, a copy of input data will get stored in the following directory:

   `SWIFT_Accelerator\SWIFT_in\IB_Archive`

For example:

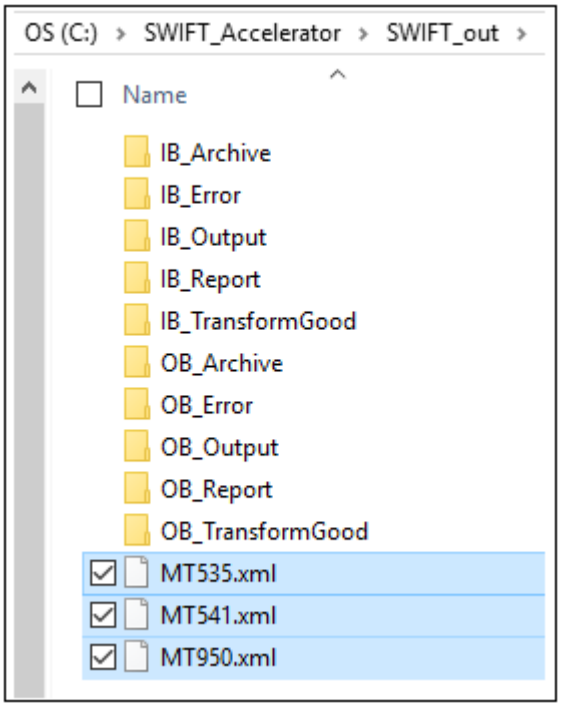


Procedure: How to Test the Sample Outbound (XML to SWIFT) Application

1. Copy the sample XML input test data to the following directory:

   `SWIFT_Accelerator\SWIFT_out`

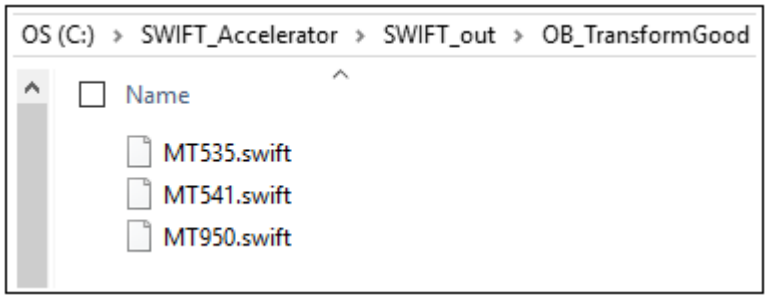
   For example:



2. Observe the transformed XML output in the following directory:

`SWIFT_Accelerator\SWIFT_out\OB_TransformGood`

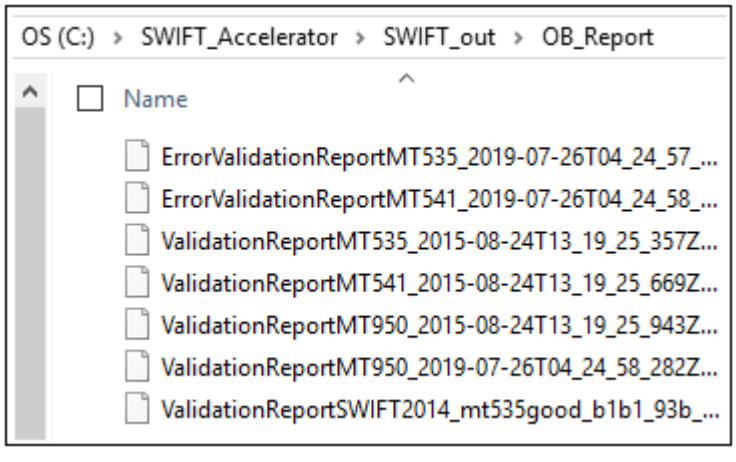
For example:



3. Observe the Reports in the following directory:

   `SWIFT_Accelerator\SWIFT_out\OB_Report`

   For example:



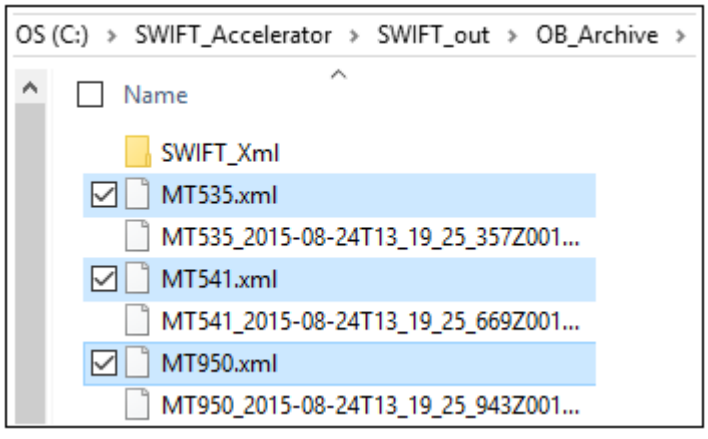
4. If any error occurs during the outbound transformation process, then you can observe error data in the following directory:

   `SWIFT_Accelerator\SWIFT_out\OB_Error`

5. After processing the input data that you place for transformation, a copy of input data will get stored in the following directory:

   `SWIFT_Accelerator\SWIFT_out\OB_Archive`

For example:


OS (C:) › SWIFT_Accelerator › SWIFT_out › OB_Archive ›
Name
SWIFT_Xml
MT535.xml
MT535_2015-08-24T13_19_25_357Z001...
MT541.xml
MT541_2015-08-24T13_19_25_669Z001...
MT950.xml
MT950_2015-08-24T13_19_25_943Z001...

Chapter 5

Inbound Processing: SWIFT to XML

The iWay Integration Solution for SWIFT includes iWay Service Manager. iWay Service Manager converts a SWIFT FIN formatted message to an XML transaction, and validates it based on published SWIFT implementation guides.

This chapter provides the information you need to understand and implement a basic inbound message flow.

- The **inbound processing overview** describes the iWay business components and the processing steps in the basic inbound message flow.
- The **sample configuration** contains the basic inbound message flow.

In this chapter:

Inbound Processing Overview

The inbound process converts a SWIFT FIN formatted message to an XML transaction.

In a basic message flow, inbound processing consists of the following components and steps. For an illustration of the components available in the construction of a message flow, see *Using a Channel to Construct a Message Flow* on page 18. You will define the components in the configuration instructions in *Sample Configuration for Inbound Processing: SWIFT to XML* on page 68.

Inlet

- The **listener** acquires the incoming SWIFT FIN message.
- The **preparser** obtains the message type from the message, in order to select the appropriate transformation template name. The transformation template and the version year from the preparser converts the original SWIFT FIN message to an XML representation of that document.

  The preparser ensures that the message is converted to a structurally correct SWIFT XML message. The transformation templates that are provided in the **Ebix** are used to transform the structure of the document.

  Set the option in the preparser to match the year of the SWIFT messages.

The iWay Integration Solution for SWIFT supports two preparsers, which are described in *Preparsers* on page 22.

Validation

- The inbound SWIFT message is validated for structure and content. The published SWIFT FIN standards and user implementation guides define element types (for example, numeric, alpha, or date) and describe rules to apply for validation.

  For example, here is a typical date segment in an inbound SWIFT message:

  ```
  :98A::STAT//20190513
  ```

  The Qualifier ("STAT") is validated against an allowed code list. The date is also validated. The tag is validated to insure correct structure.

Route

- After validation, you can apply any additional business logic to the document. You can use a single service or multiple services, passing the output of one service to the input of the next.

  In the sample inbound channel, a process flow is used to apply specific logic.

  For more information on available services, see the *iWay Service Manager User's Guide*.

Outlets

Outlets define how messages leave a channel at the end of a process. In the sample inbound channel, a *Passthrough* outlet is used. All write operations are handled in the process flow.

Sample Configuration for Inbound Processing: SWIFT to XML

This topic highlights the configuration of the sample inbound message flow for the iWay Integration Solution for SWIFT. The message flow represents the movement and tasks in the conversion of a message from SWIFT FIN format to XML format.

Adding an Ebix or Registers to the Application Project

The iWay e-Business Information Exchange (Ebix) framework supplies several Ebix files for the iWay Integration Solution for SWIFT.

An Ebix file for SWIFT is named SWIFT_*ccyy*.ebx, where *ccyy* is the release year. For example, an Ebix file for the 2019 SWIFT FIN messages is named SWIFT_2019.ebx.

For more information on the supported SWIFT FIN messages, see *Ebix-Supported Transaction Sets* on page 89.

Note: You can download the latest Ebix archives from the Information Builders Technical Support Center:

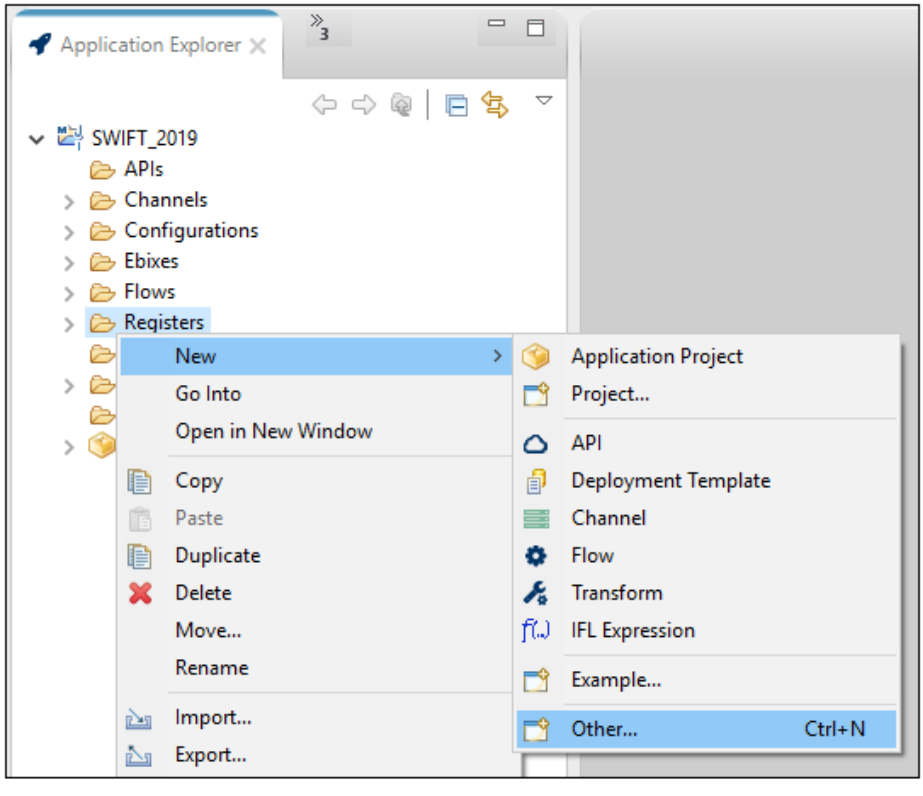
https://techsupport.informationbuilders.com/

Note: The SWIFT 2019 Ebix is attached to the sample channels.

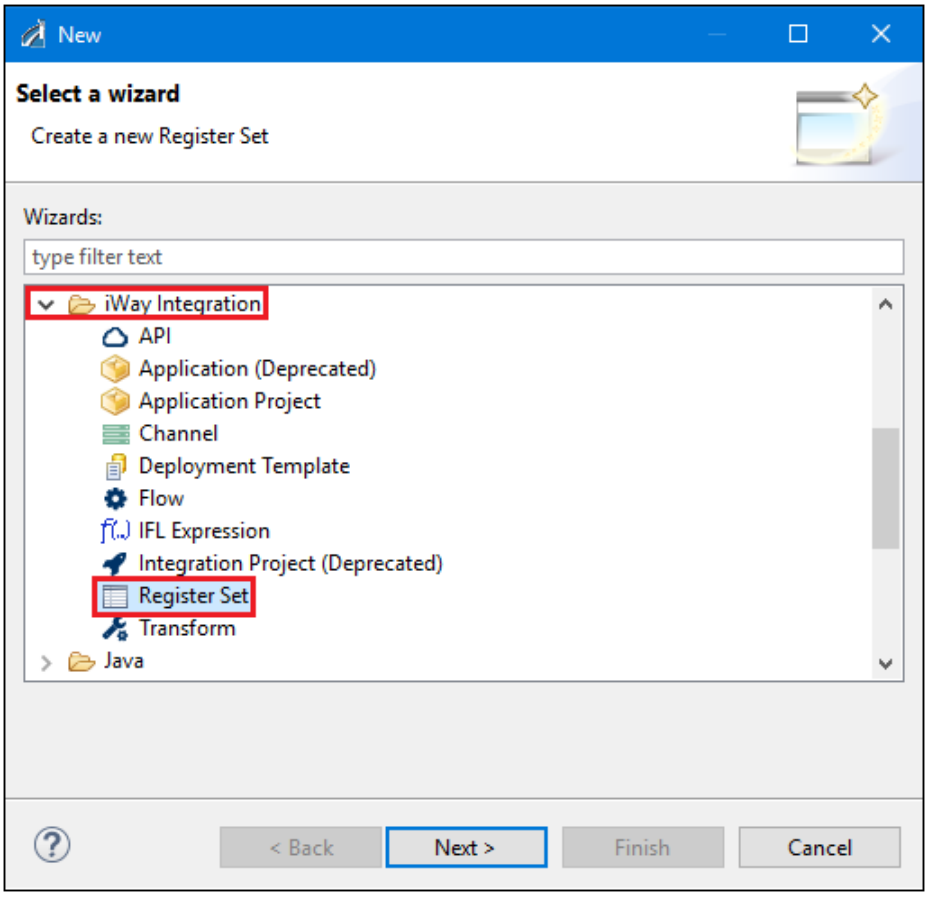
For more information on how to add (import) an Ebix to an application project, see *How to Import an Ebix* on page 93.

Procedure: How to Add Registers to the Application Project

1. Right-click the *Registers* subfolder in your application project, select *New*, and then click *Other* from the context menu, as shown in the following image.

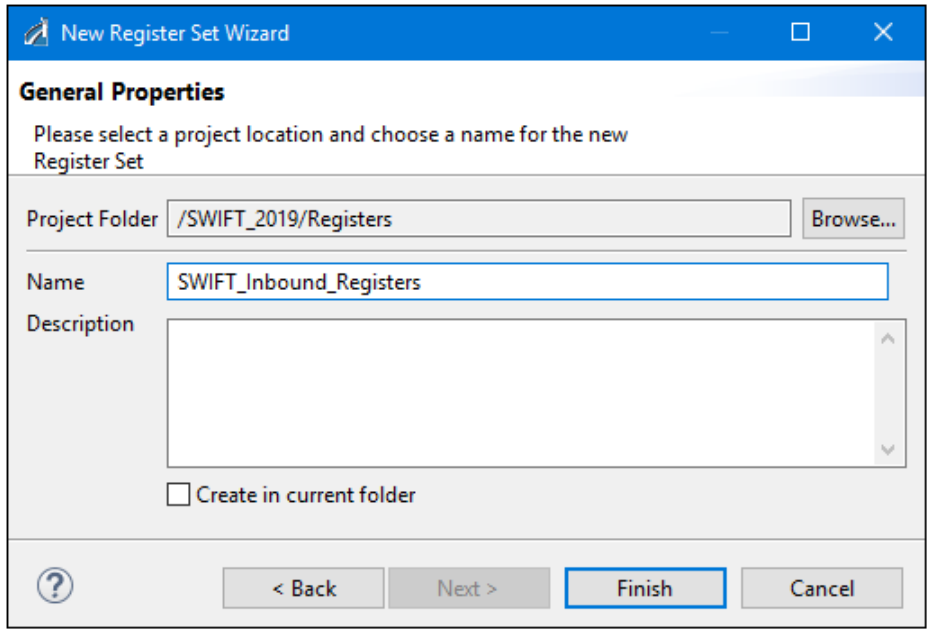


The New (Select a wizard) dialog opens, as shown in the following image.



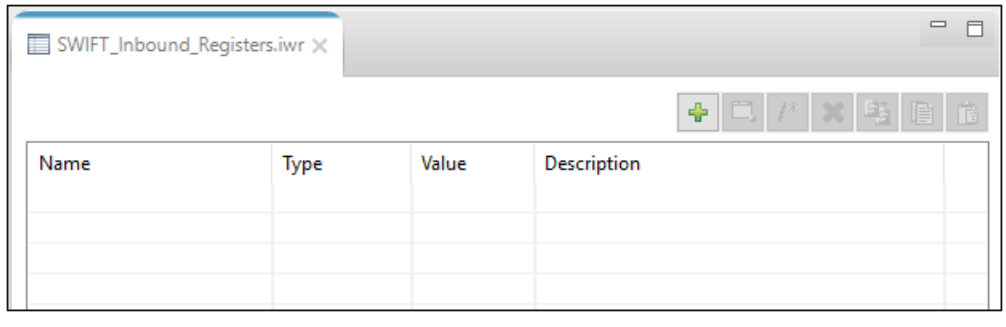
2. Expand *iWay Integration*, select *Register Set*, and then click *Next*.

The New Register Set Wizard opens, as shown in the following image.

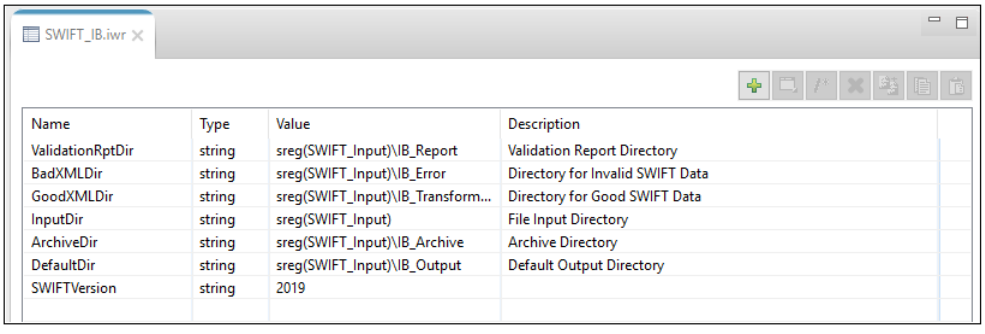


3. Enter a name for the register set and then click *Finish*.

   The named register set opens as a new tab in the workspace, where you can define your registers, as shown in the following image.



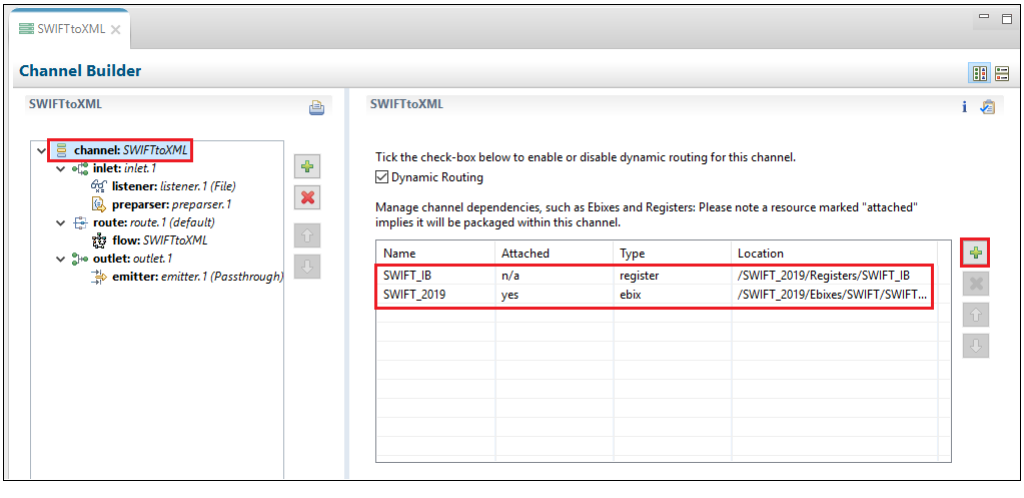
The following image shows a sample register set (SWIFT_IB), which is used for inbound processing.



Procedure: How to Add an Ebix and Registers to Your Channel

1. Expand the *Channels* subfolder in your application project, and double-click on a channel name.

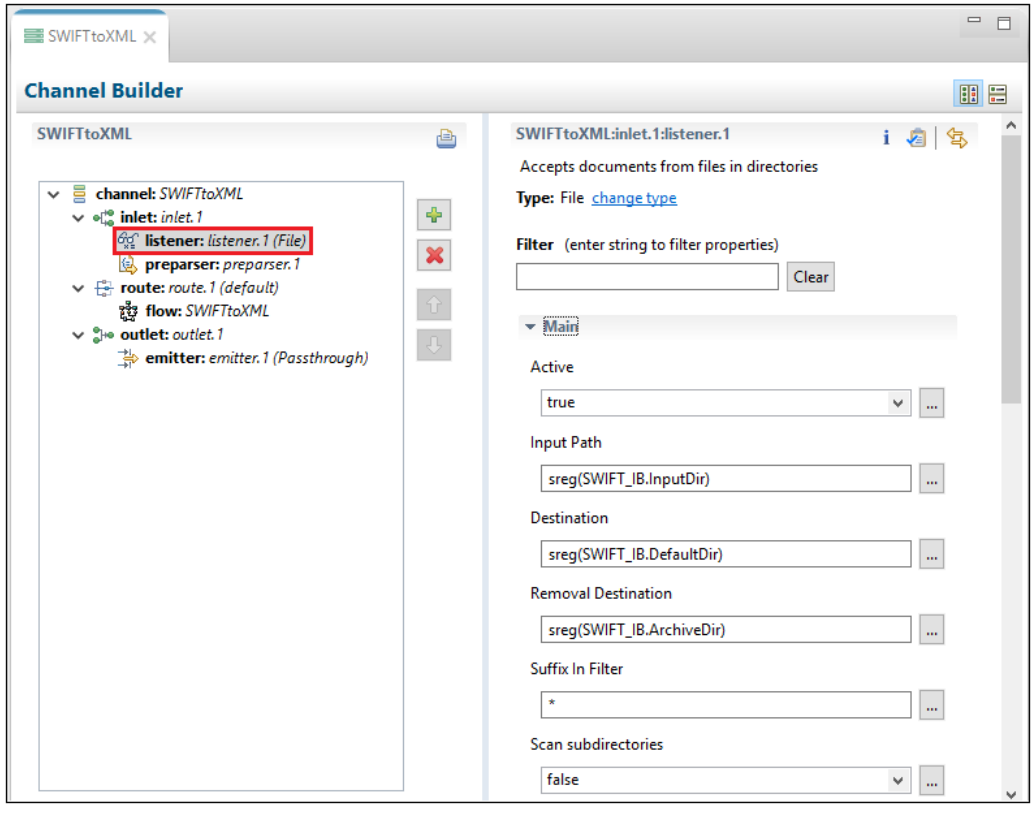
   The selected channel opens in the Channel Builder, which displays as a tab in your workspace, as shown in the following image.



2. Click on the channel name in the left pane of the Channel Builder to open the dialog in the right pane, which allows you to bind Ebixes and registers to your channel.
3. Click the *Add* button (green plus sign) and add your Ebix or register sets that were defined.

Sample Listener Configuration

The following image shows a sample listener configuration for the inbound channel.



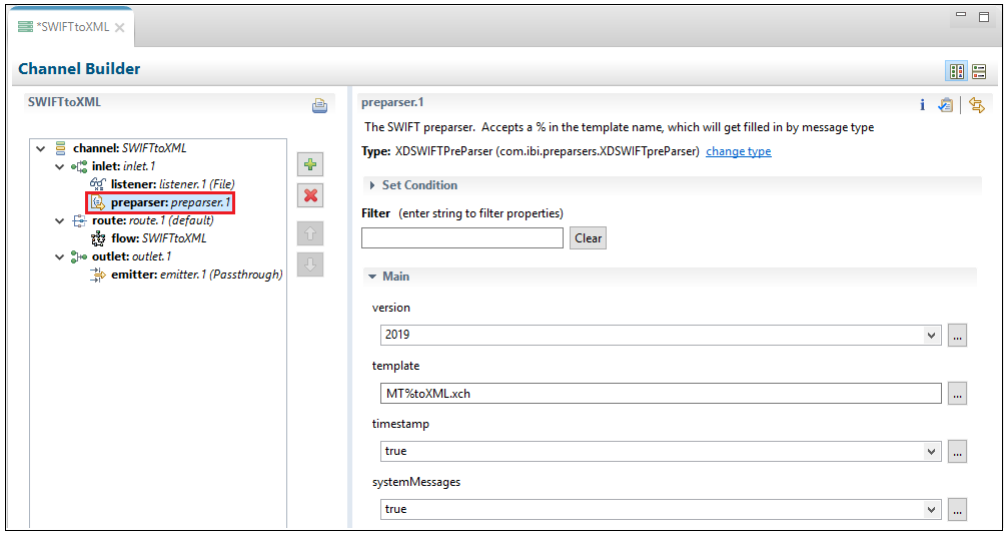
The following table lists and describes the available configuration parameters for the listener.

Parameter	Value
Active	Enables (set to *true* by default) or disables the listener.

Parameter	Value
Input Path	sreg(SWIFT_IB.InputDir) This value is a special register that uses a defined directory in which input messages are received. Ensure that you have created this directory; otherwise, errors will occur during deployment.
Destination	sreg(SWIFT_IB.DefaultDir) This value is a special register that uses a defined directory in which output files are stored after transformation. Ensure that you have created this directory; otherwise, errors will occur during deployment.
Removal Destination	sreg(SWIFT_IB.ArchiveDir) This value is a special register that uses a defined directory to which input messages are moved if they fail during transformation. Ensure that you have created this directory; otherwise, errors will occur during deployment. It is recommended to configure a removal destination when you are constructing a basic channel.
Suffix In Filter	* Input files with any file extension are allowed.
Suffix Out	xml The extension for output files is .xml.
Accept Zero Length Files?	true If true, listener expects flat (non-XML). Automatic parsing is not performed.

Sample Preparser Configuration

The following image shows a sample preparser configuration for the inbound channel.



The following table lists and describes the available configuration parameters for the preparser:

Parameter	Description
version	Represents the SWIFT release year you are using.
template	Represents the template to use for the SWIFT to XML transform. Enter the following: `MT%toXML.xch`
timestamp	Writes a time stamp to the log file. Select *true* or *false*.

Parameter	Description
systemMessages	Determines whether to process system messages. If *true* is selected, the system messages are automatically parsed and directly written to a good validation report. If *false* is selected, the system messages are bypassed and not processed. There is also a separate preparser available if you want to set up a standalone channel to parse **just** system messages.

Defining a Route

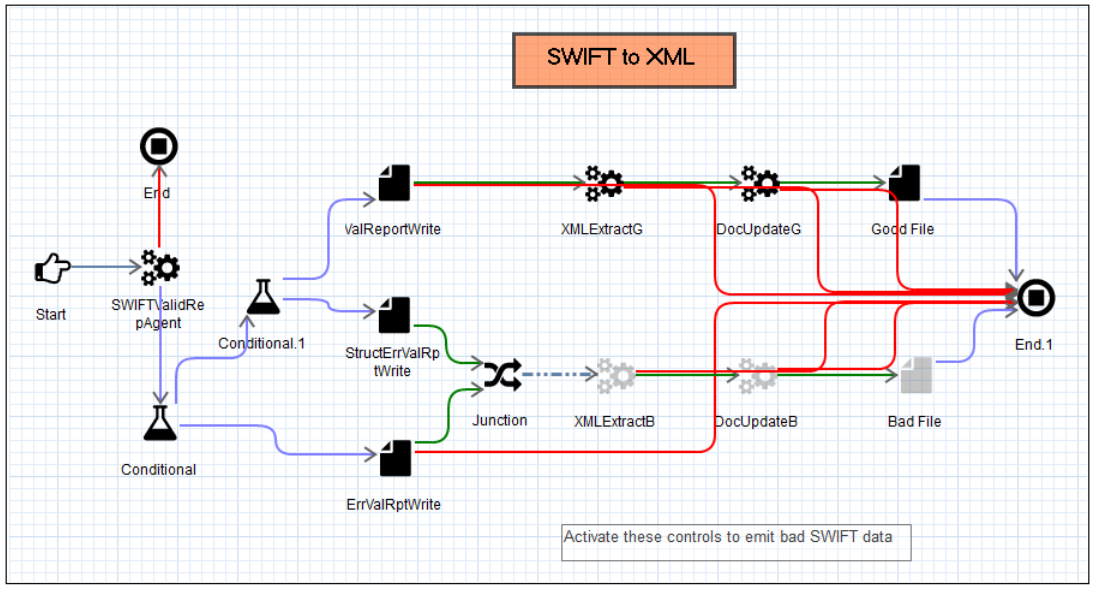
A route defines the path a message takes through a channel.

Channel Builder defines the route for you automatically. Just add your process flow(s) to the route in Channel Builder.

Defining a Process Flow

Process flows (pflows) are defined and created in the process flow editor. To create a new process flow, right-click on *Flows*, select *New*, and then click *Flows*, and follow the dialog.

The following image is a sample process flow for the inbound channel.



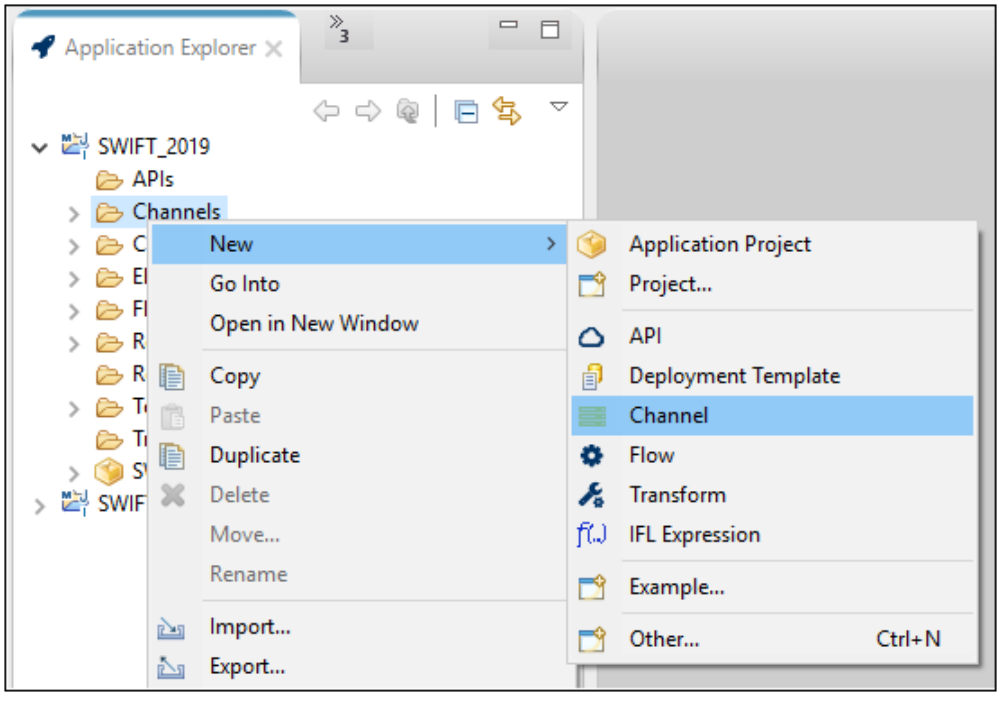
Defining the Outlet

Outlets define how a message leaves the channel. In the sample inbound configuration, a *Passthrough* outlet is used. For details on supported protocols, which can be used as outlets, see the *iWay Service Manager Protocol Guide*.

Procedure: How to Build the Channel

To build the channel:

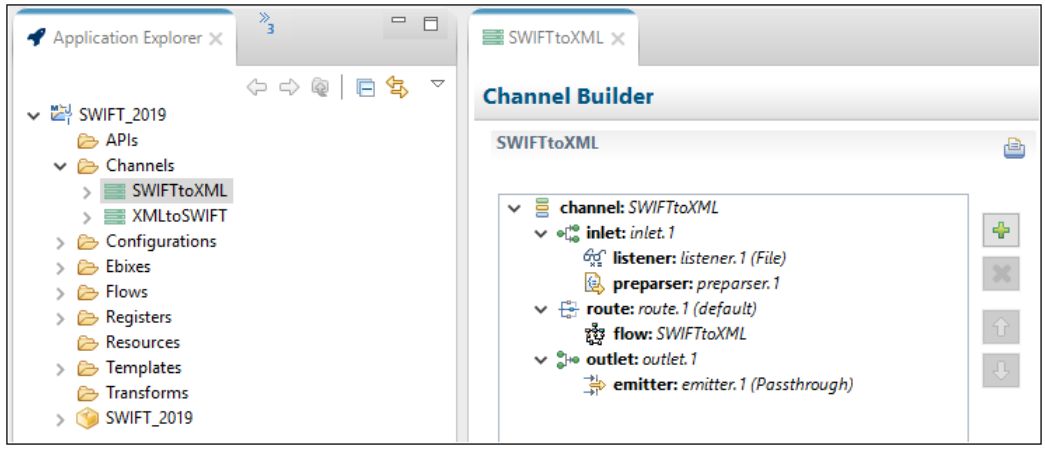
1. Expand your application project, right-click *Channels*, select *New*, and then click *Channel* from the context menu, as shown in the following image.



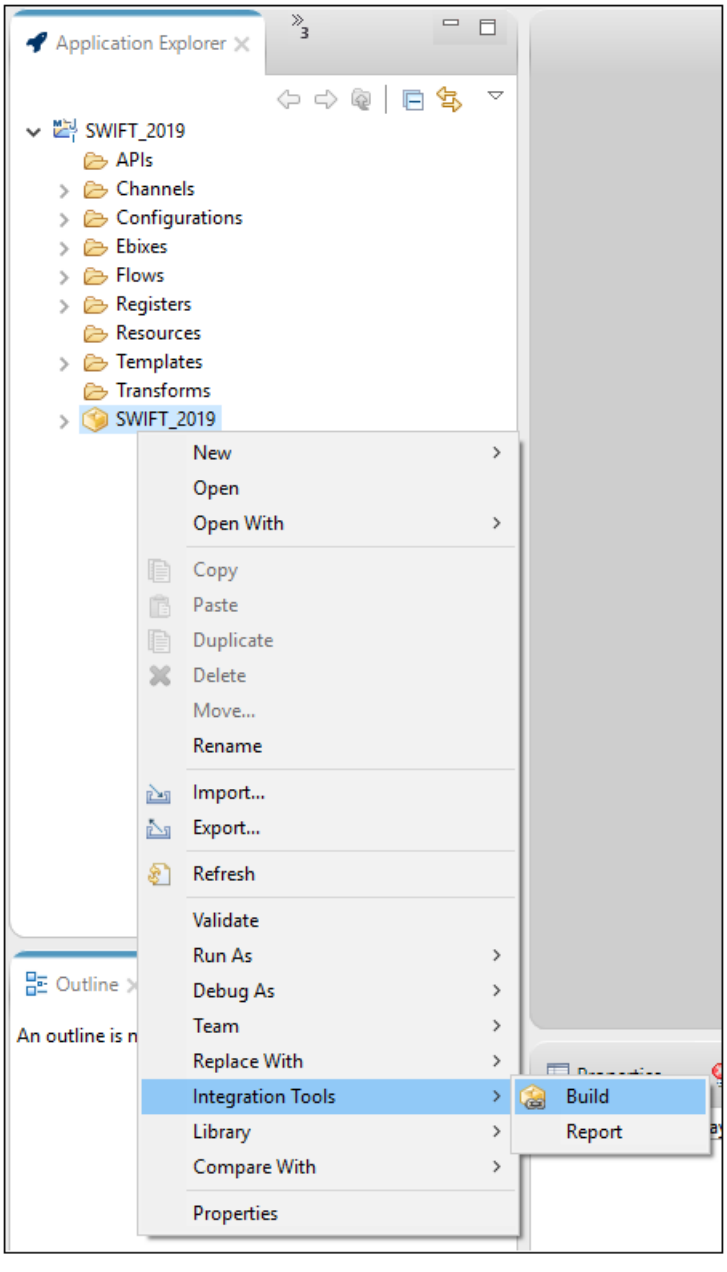
   The Channel Object dialog opens.

2. Enter a new channel name and then click *Next* to continue to be prompted for entry screens for components, or click *Finish* to go directly to the Channel Builder.

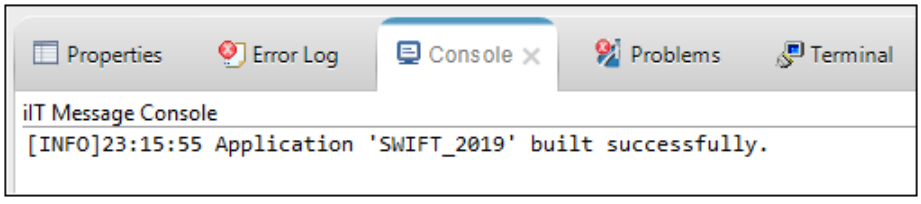
The following image shows a sample inbound channel displayed in the Channel Builder.



You can manually build your channel by right-clicking the bundle name, selecting *Integration Tools*, and then clicking *Build* from the context menu, as shown in the following image.



3. Review the results of your build, as shown in the following image.



Deploying and Starting the iWay Integration Application

For more information, see:

- *Deploying the iWay Integration Application for SWIFT* on page 52
- *Starting iWay Integration Applications in iWay Service Manager* on page 57

Procedure: How to Verify the Channel

To ensure that the channel is working as expected, perform the following steps:

1. Place a SWIFT document as test data in the file directory that is defined by *sreg(SWIFT_IB.InputDir)*.

   For more information on obtaining SWIFT sample files (MT535.swift, MT541.swift, and MT950.swift) for testing purposes, see *Extracting SWIFT User Directories and Data Samples* on page 42.

2. Check for the XML file and the validation report in the file directory that is defined by *sreg(SWIFT_IB.DefaultDir)*. This is the destination path you specified for the emitters associated with the outlets for the channel.

   For example, a SWIFT input file named MT950.txt is named MT950_2019-11-27T17_17_38_932Z.xml on output.

Chapter 6

Outbound Processing: XML to SWIFT

The iWay Integration Solution for SWIFT validates an XML document based on published implementation guides for SWIFT and converts it to a document in SWIFT format.

This chapter provides the information you need to understand and implement a basic outbound message flow.

- The **outbound processing overview** describes the iWay business components and the processing steps in the basic outbound message flow.
- The **sample configuration** shows a basic outbound message flow.

In this chapter:

Outbound Processing Overview

The standard outbound process converts an XML transaction to a SWIFT FIN message.

In a basic message flow, outbound processing consists of the following components and steps. For an illustration of the components available in the construction of a message flow, see *Using a Channel to Construct a Message Flow* on page 18.

Inlet

- The **listener** acquires the input XML document.

Route

The following services are defined:

- **SWIFT XML Transform service**, which is used to transform SWIFT XML to SWIFT messages.
- **Validation Report service**, which is used to validate SWIFT messages against SWIFT defined standards.

Using iWay Integration Tools (iIT) Designer, a process flow is created which incorporates these two services.

Outlets

Outlets define how messages leave a channel at the end of a process. In the sample outbound channel, a *Passthrough* outlet is used. All write operations are handled by the process flow.

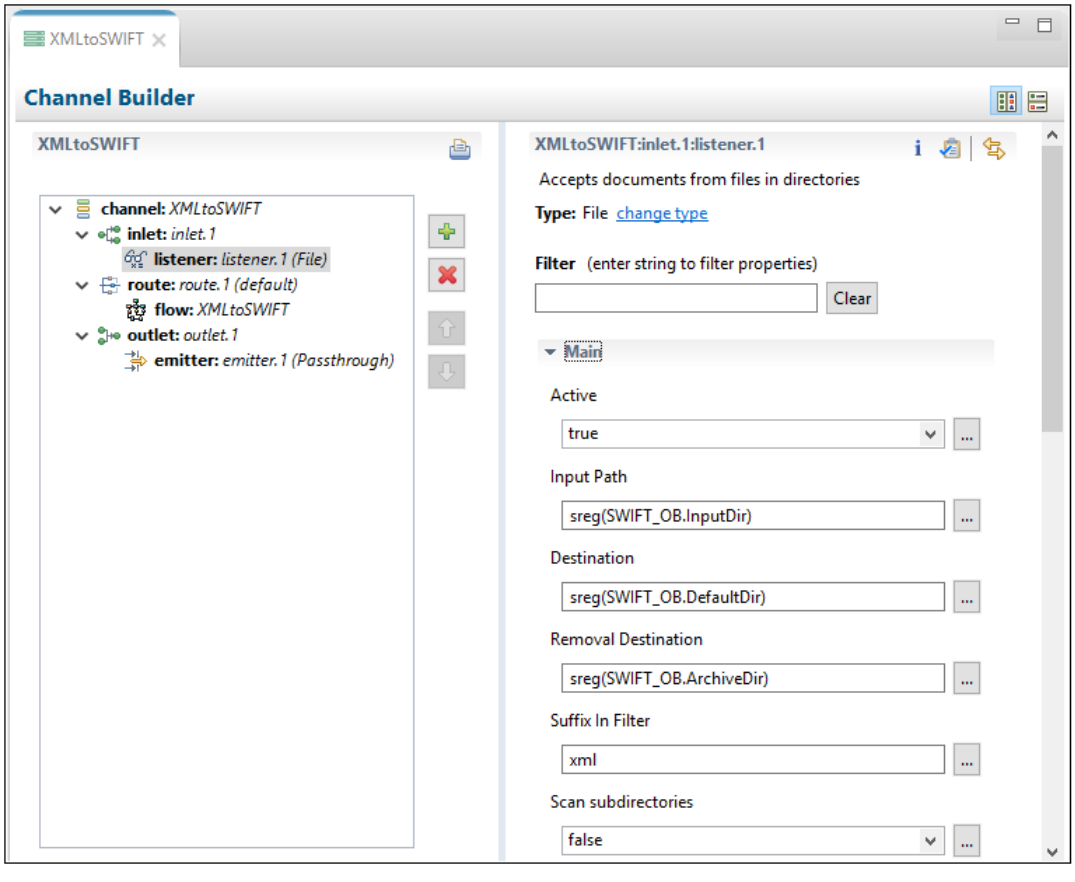
Adding an Ebix to the Registry

For more information on adding an Ebix to the registry, see *How to Add an Ebix and Registers to Your Channel* on page 72.

Tip: If you already added an Ebix to the application project for inbound processing, then you do not need to add it again for outbound processing. as described in *How to Add an Ebix and Registers to Your Channel* on page 72.

Sample Listener Configuration

The following image shows a sample listener configuration for the outbound channel.

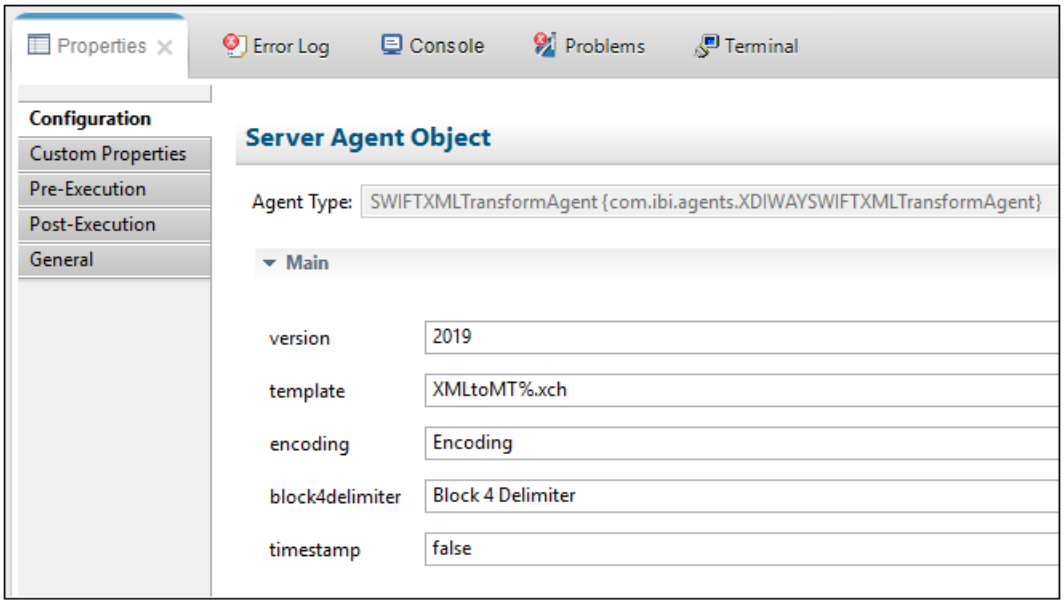


The following table lists and describes the configuration parameters for the sample listener.

Parameter	Value
Active	Enables (set to *true* by default) or disables the listener.
Input Path *	sreg(SWIFT_OB.InputDir) This value is a special register that uses a defined directory in which input messages are received. Ensure that you have created this directory; otherwise, errors will occur during deployment.
Destination *	sreg(SWIFT_OB.DefaultDir) This value is a special register that uses a defined directory in which output files are stored after transformation. Ensure that you have created this directory; otherwise, errors will occur during deployment.
Removal Destination	sreg(SWIFT_OB.ArchiveDir) This value is a special register that uses a defined directory to which input messages are moved if they fail during transformation. Ensure that you have created this directory; otherwise, errors will occur during deployment. It is recommended to configure a removal destination when you are constructing a basic channel.
Suffix In Filter	xml Input files with the extension .xml are allowed.
Suffix Out	swift In this example, the extension for output files is .swift.

Sample Transformation Agent Properties

The following image shows a sample transformation agent.



The following table lists and describes the available configuration parameters for the preparser.

Parameter	Description
version	Represents the SWIFT release year you are using.
template	Represents the template to use for the XML to SWIFT transform. Enter the following: `XMLtoMT%.xch`
encoding	Determines how to encode the output.
block4delimiter	Inserts a delimiter into SWIFT Block 4.
timestamp	Writes a time stamp to the log file. Select *true* or *false*.

Defining a Route

A route defines the path a message takes through a channel.

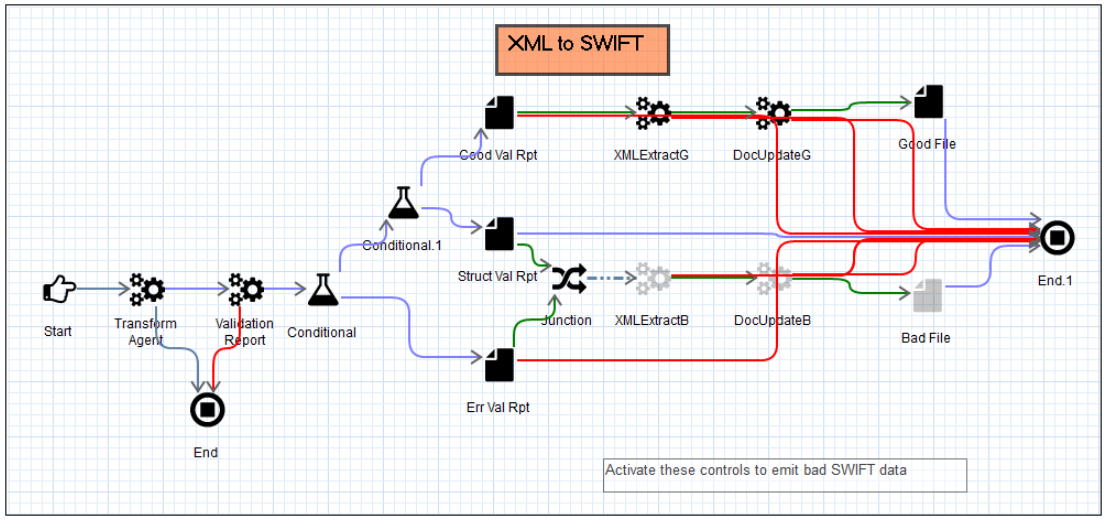
Channel builder defines the route for you automatically. Add your process flow(s) to the route in Channel Builder.

Defining a Process Flow

Process flows (pflow) are defined and created in the process flow editor.

To create a new process flow, right-click on *Flows*, select *New*, then click *Flows*, and follow the dialog.

The following image shows a sample process flow for the outbound channel.



Defining the Outlet

Outlets define how a message leaves a channel. In the sample inbound configuration, a *Passthrough* outlet is used. For details on supported protocols, which can be used as outlets, see the *iWay Service Manager Protocol Guide*.

Adding the Ebix to the Channel

For more information on adding an Ebix to the Channel, see *How to Add an Ebix and Registers to Your Channel* on page 72. If one Ebix is shared by both inbound and outbound channels, it must be added to each channel.

Building the Channel

For more information on building the channel, see *How to Build the Channel* on page 78.

Deploying and Starting the iWay Integration Application

For more information, see:

- *Deploying the iWay Integration Application for SWIFT* on page 52
- *Starting iWay Integration Applications in iWay Service Manager* on page 57

Note that an iWay Integration Application (iIA) may contain multiple channels. The entire iIA is deployed and each channel is started individually.

Procedure: How to Verify the Channel

To ensure that the channel is working as expected, perform the following steps:

1. Place an XML file as test data in the file directory that is defined by *sreg(SWIFT_OB.InputDir)*.

   For more information on obtaining sample XML input files (MT535.xml, MT541.xml, and MT950.xml) for testing purposes, see *Extracting SWIFT User Directories and Data Samples* on page 42.

2. Check for the SWIFT formatted output file in the file directory that is defined by *sreg(SWIFT_OB.DefaultDir)*. This is the destination directory you specified.
3. Confirm that the output has been converted to SWIFT format.

Appendix A

Ebix-Supported Transaction Sets

This topic describes the SWIFT FIN messages supported by the iWay Integration Solution for SWIFT in the Ebix files supplied with the product.

In this appendix:

Transaction Set Support

iWay Service Manager (iSM) Version 8.0 and higher supports SWIFT 2016, SWIFT 2017, SWIFT 2018, and SWIFT 2019 . Note that many documents did not have any updates in SWIFT 2019, and some remain unchanged since SWIFT 2008. For additional information and available Ebix file downloads, go to *http://techsupport.ibi.com*.

SWIFT Version	Documents
2016	All
2017	All
2018	All
2019	All

Appendix B

Using iWay Integration Tools to Configure an Ebix for SWIFT

This section describes how to use iWay Integration Tools (iIT) to configure an e-Business Information Exchange (Ebix) file for SWIFT.

In this appendix:

Overview

You can use iWay Integration Tools (iIT) to import, edit, export, and work with e-Business Information Exchange (Ebix) files for SWIFT. The topics in this appendix describe how to:

- Import a SWIFT 2019 MT535 Ebix into iIT.
- Add a qualifier at the 19A element under the L_19A loop level to the SWIFT Ebix.
- Export the edited Ebix to a physical location.

The edited Ebix can be returned and then tested with the appropriate SWIFT 2019 MT535 message.

Prerequisites

Before you continue, ensure that the following prerequisites are met:

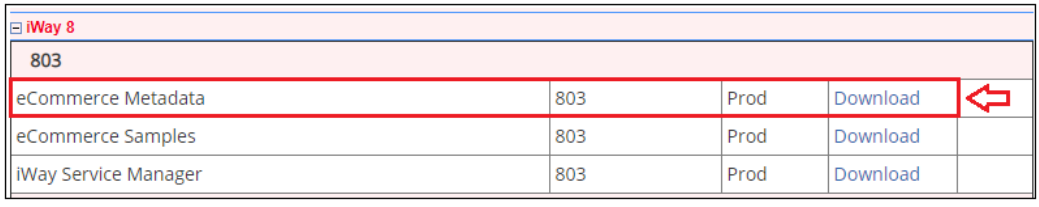
- You have a working knowledge of iWay Integration Tools (iIT) and SWIFT.
- iWay E-Business Adapter for SWIFT is installed.
- iIT Version 8.0 or higher is installed.

Downloading and Extracting Ebix Files For SWIFT

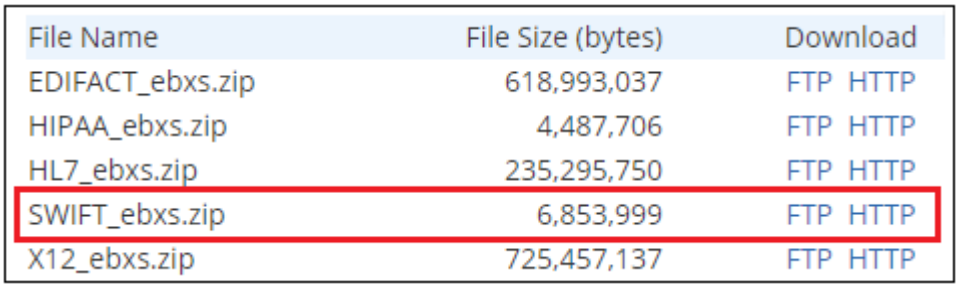
To download and extract Ebix files for SWIFT:

1. Download the eCommerce Metadata samples for SWIFT (version 8.0.3) from the Information Builders Technical Support Center:

https://techsupport.informationbuilders.com/

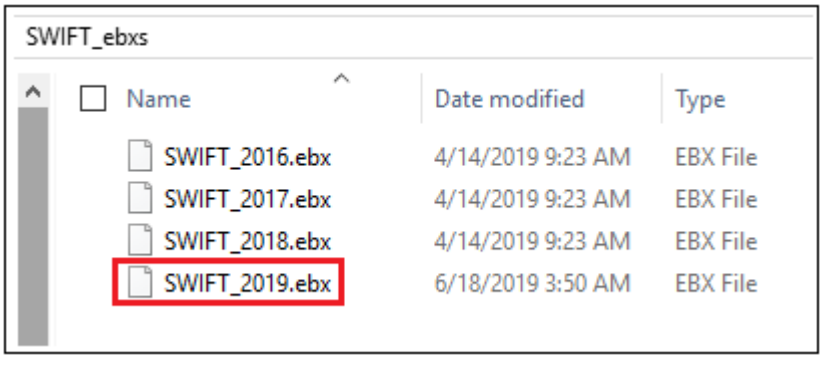


2. Download the *SWIFT_ebxs.zip* file, as shown in the following image.



3. Unzip the downloaded *SWIFT_ebxs.zip* file into any physical location on your local drive.

   When extracted, the *SWIFT_ebxs* folder is created, which contains individual Ebix files for several SWIFT versions, as shown in the following image.



   For example, the *SWIFT_2019.ebx* file contains the SWIFT 2019 MT535 document.

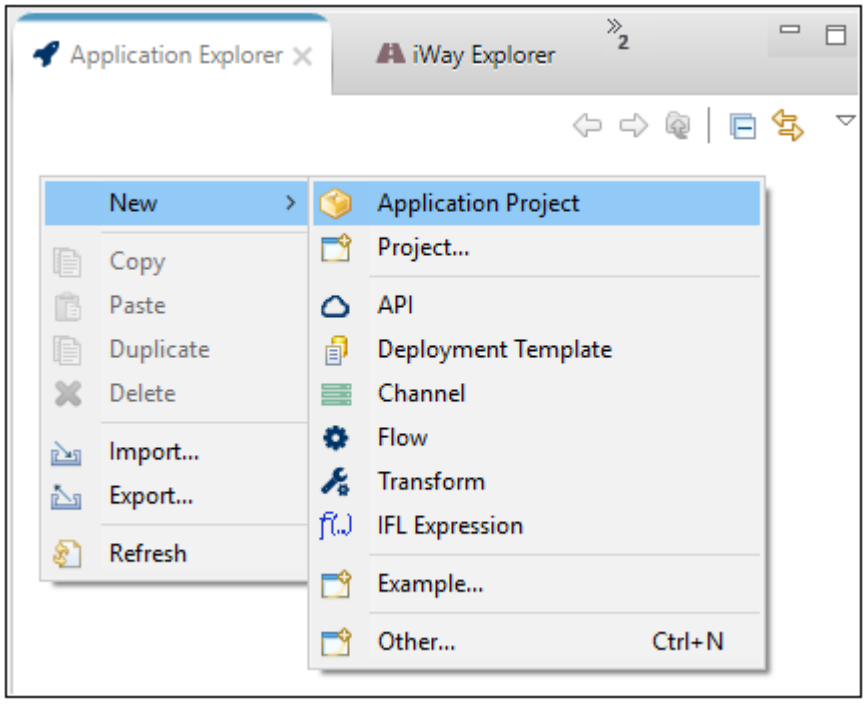
   Note: Ensure all folders used for the extracted *SWIFT_ebix.zip* file do not have any blank spaces in the folder name.

Working With Ebix Files Using iWay Integration Tools

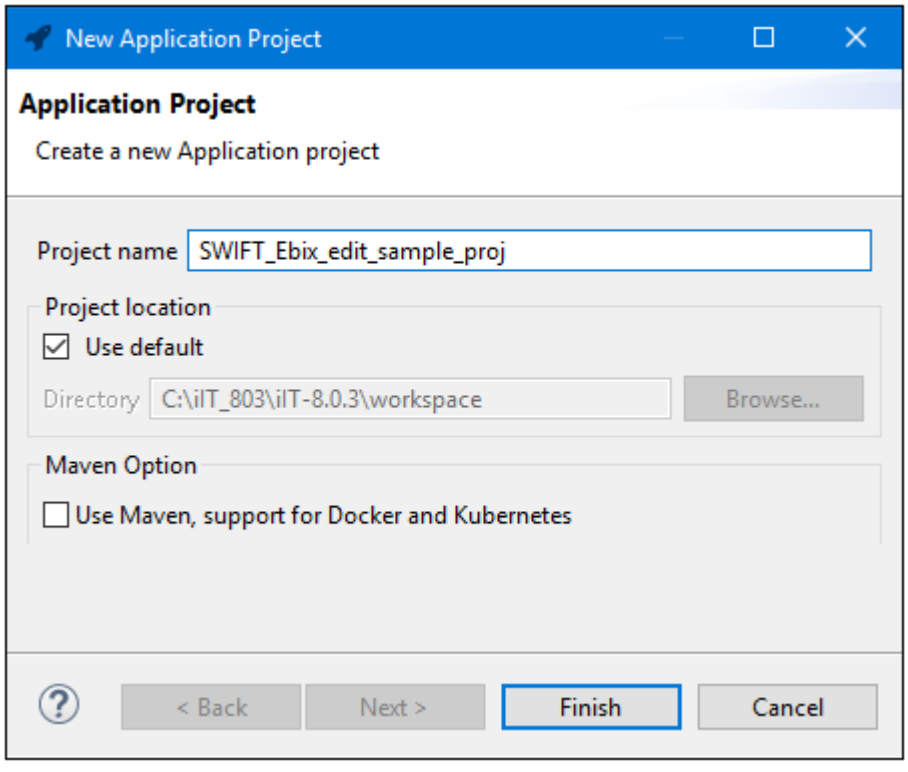
This section describes how to import, edit, and export an Ebix for SWIFT using iWay Integration Tools (iIT).

Procedure: How to Import an Ebix

1. Start iWay Integration Tools (iIT).
2. Right-click anywhere in the Application Explorer tab, click *New*, and then select *Application Project* from the context menu, as shown in the following image.

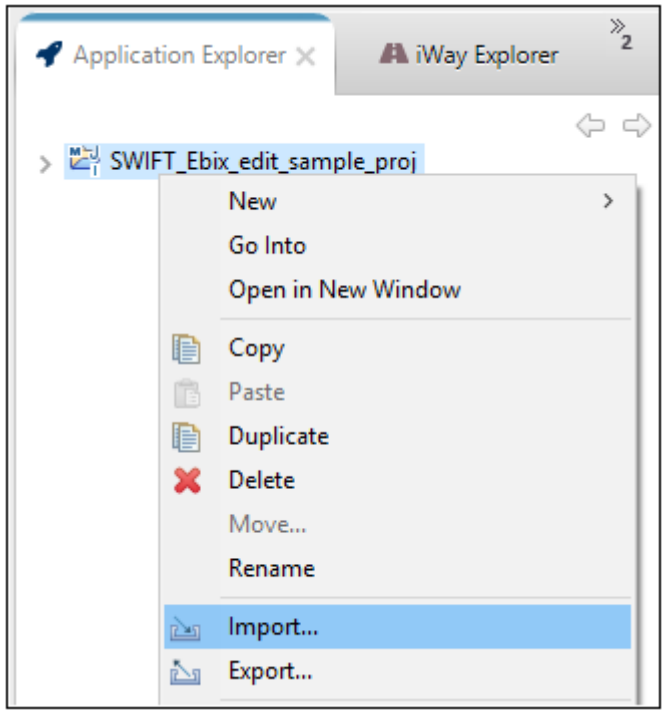


The New Application Project dialog opens, as shown in the following image.



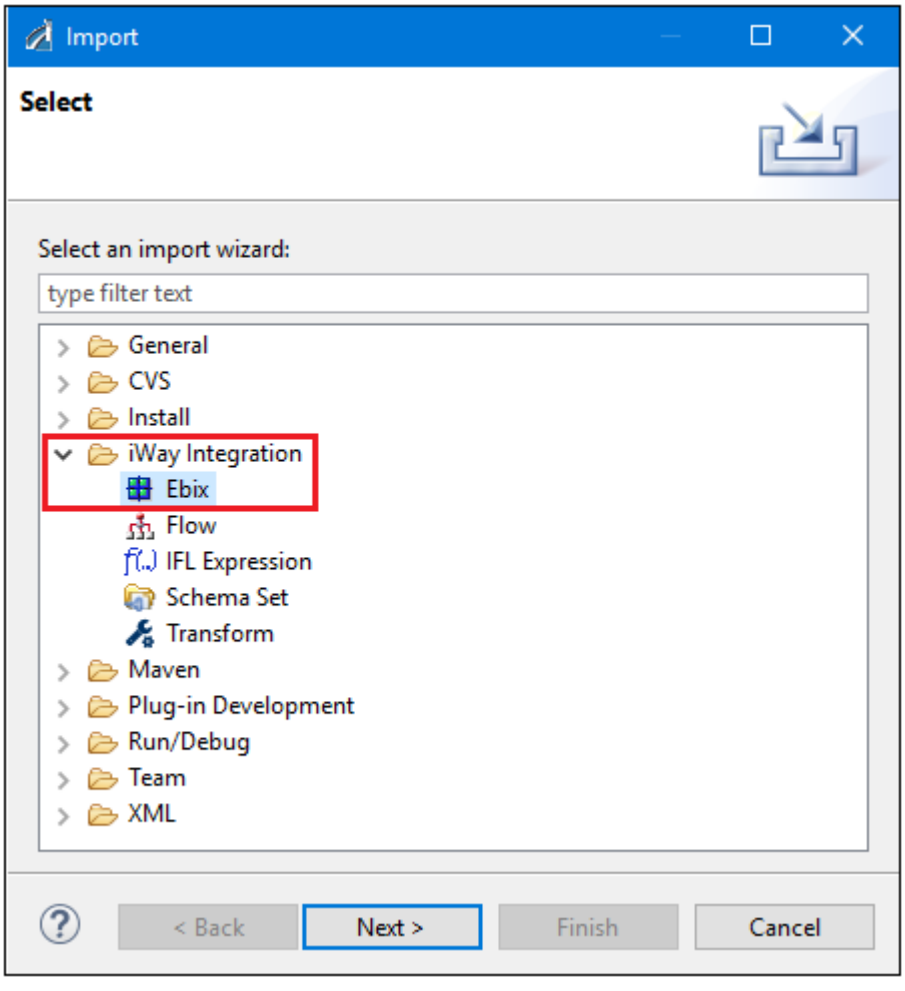
3. Enter a project name, for example, *SWIFT_Ebix_edit_sample_proj*, and then click *Finish*.

Your new project is created and is listed in the Application Explorer tab, as shown in the following image.



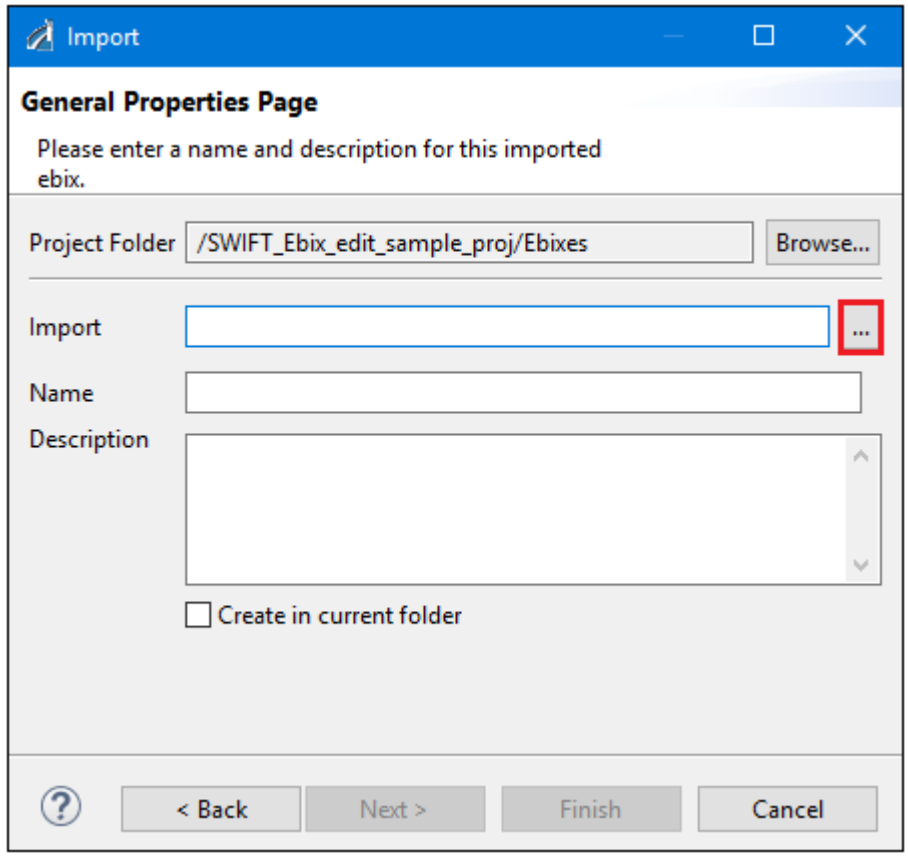
4. Right-click the project in the Application Explorer tab and select *Import* from the context menu.

The Import dialog opens, as shown in the following image.



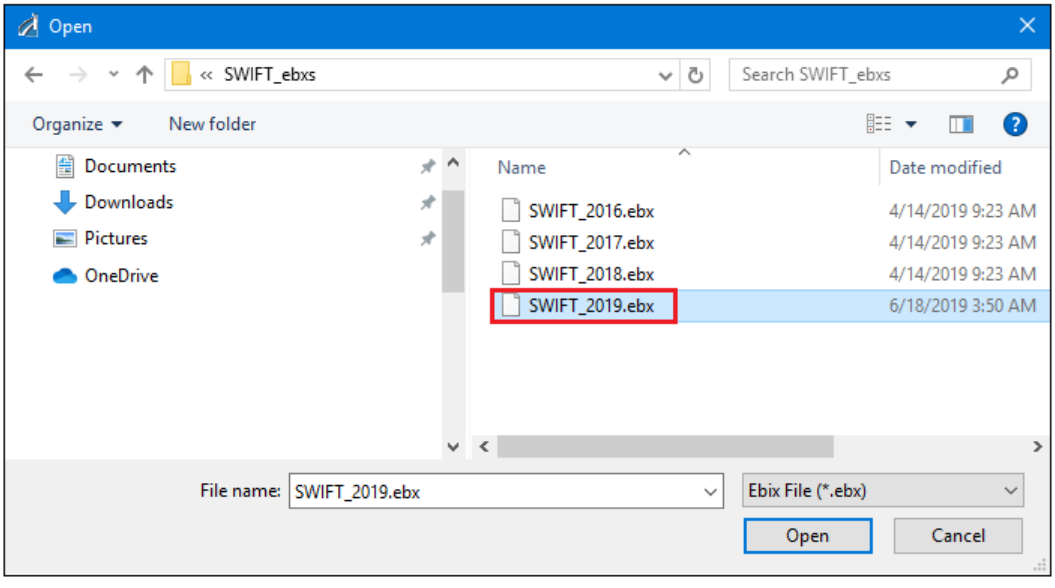
5. Expand *iWay Integration*, select *Ebix*, and then click *Next*.

The General Properties Page opens, as shown in the following image.



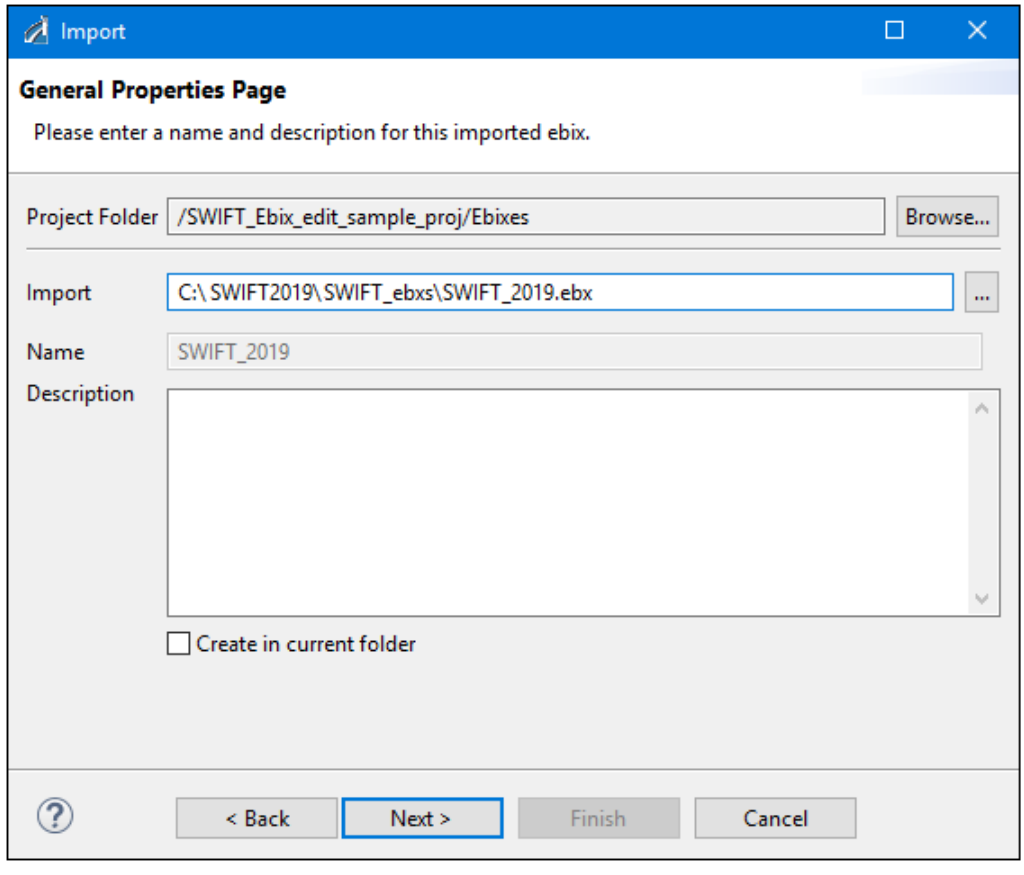
6. Click the *ellipsis* (…) button to the right of the Import field.

The Open dialog is displayed, as shown in the following image.



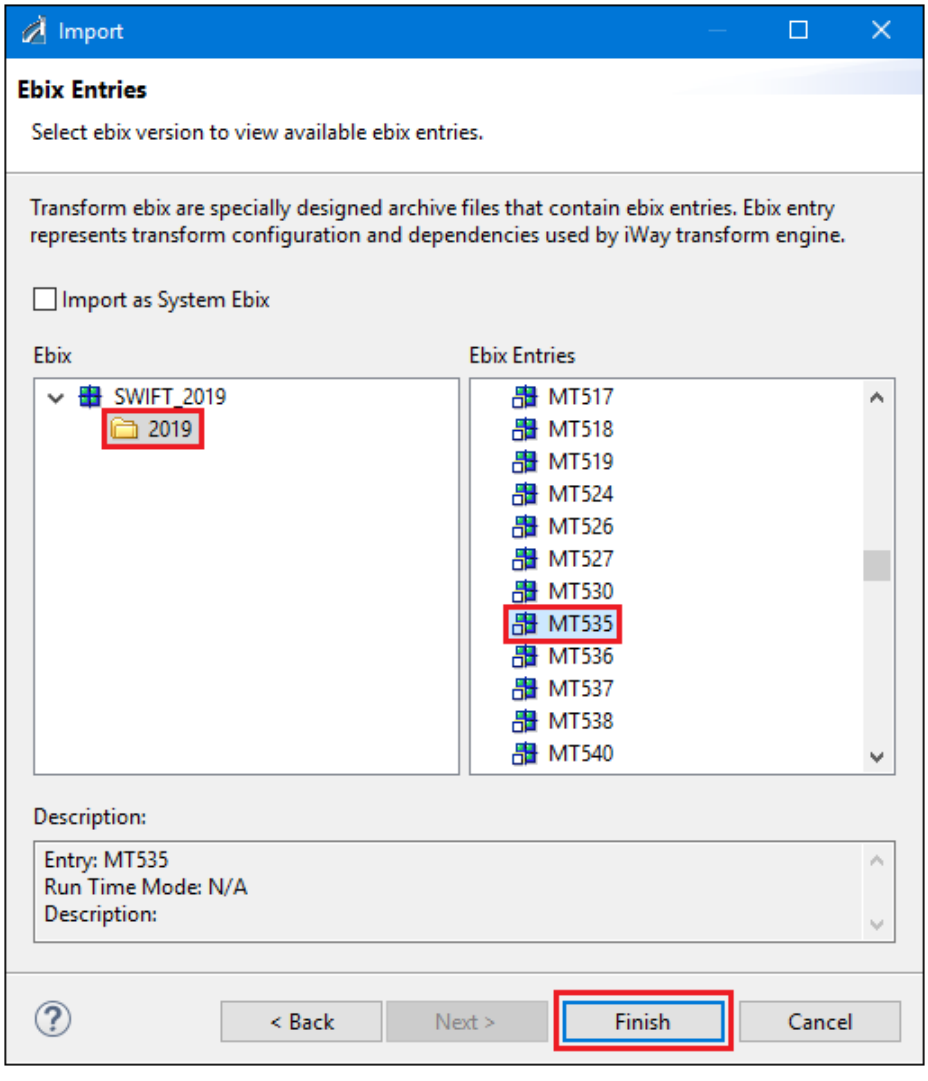
7. Select the downloaded *SWIFT_2019.ebx* file from the physical drive location, and then click *Open*.

You are returned to the General Properties Page, as shown in the following image.



8. Click *Next*.

The Ebix Entries pane opens, as shown in the following image.

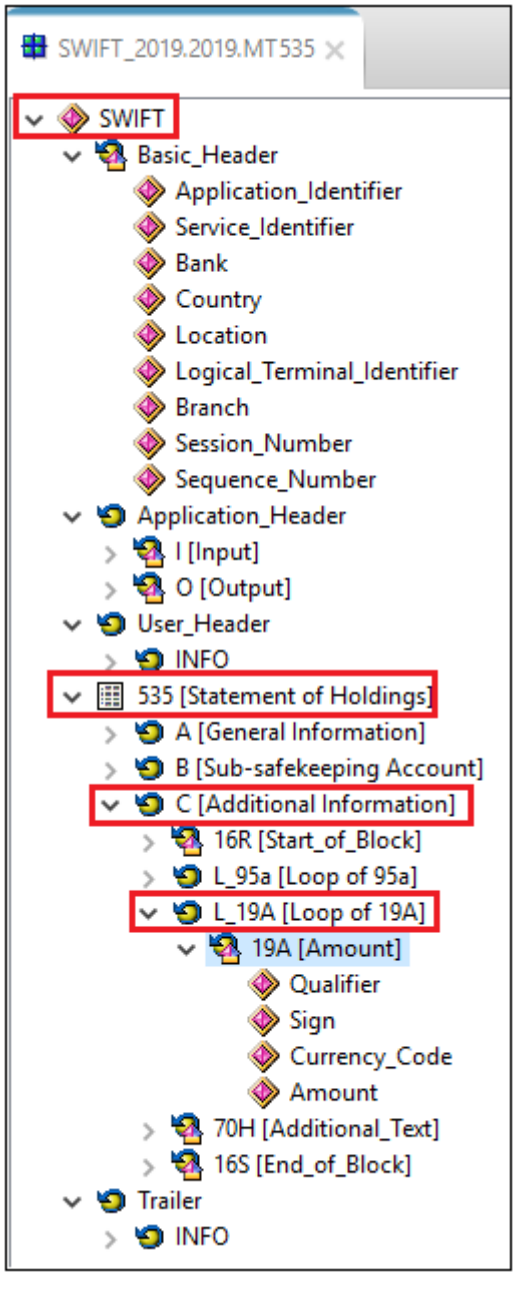


9. Expand the *SWIFT_2019* node in the Ebix section on the left, and select the *2019* folder.
10. Select *MT535* in the Ebix Entries section on the right, and then click *Finish*.

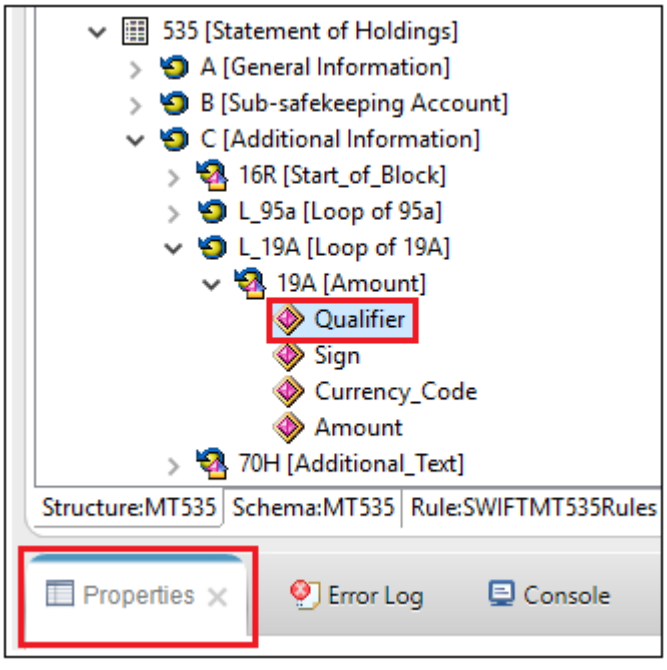
    The dictionary editor opens for your selected document (for example, MT535).

Procedure: How to Edit an Ebix

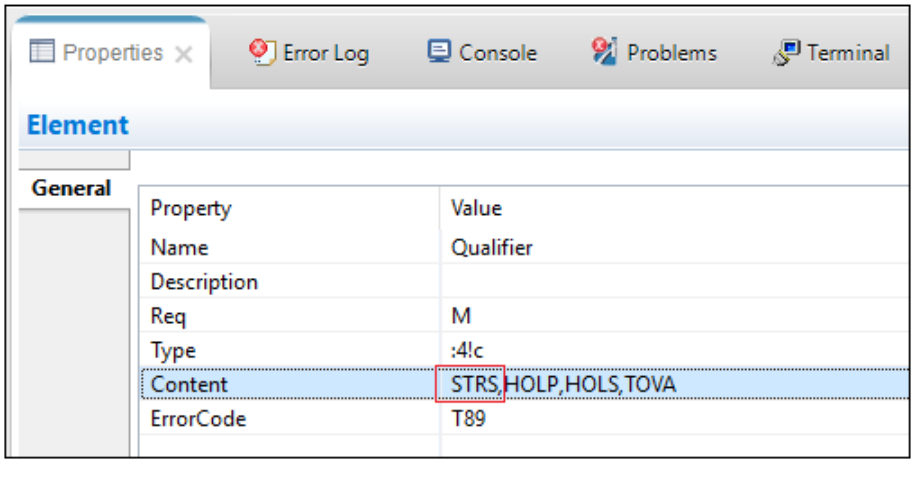
1. In the *SWIFT_2019.2019.MT535* tab that is opened, navigate to the *19A [Amount]* element by expanding *SWIFT*, *535 [Statement of Holdings]*, *C [Additional Information]*, and then *L_19A [Loop of 19A]*, as shown in the following image.



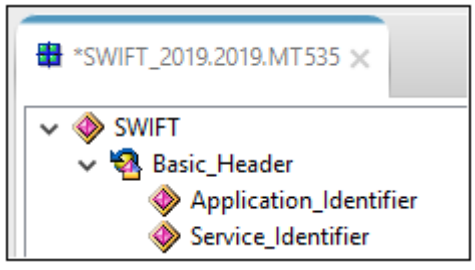
2. Under *19A [Amount]*, click the *Qualifier* composite element, then click the *Properties* tab in the bottom pane, as shown in the following image.



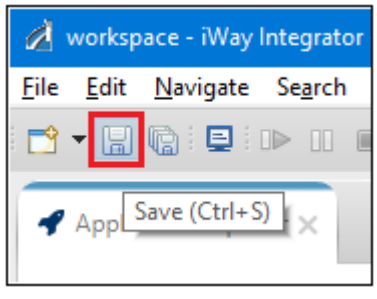
3. In the Content field of the Properties tab, add the value *STRS,* as shown in the following image.



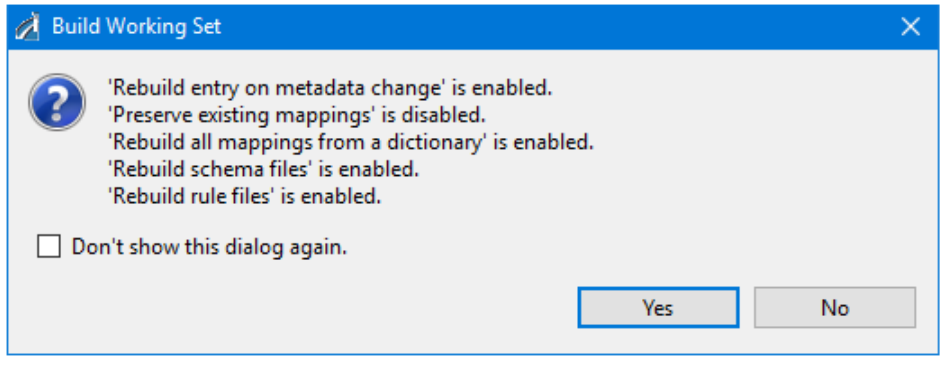
An asterisk (*) character appears next to the file name in the tab until you have saved the edited changes, as shown in the following image.



4. Save your edited Ebix by clicking the *Save* icon, which is located on the toolbar. If you are using a Windows platform, you can also use the shortcut key *CTRL*+S to save your work.

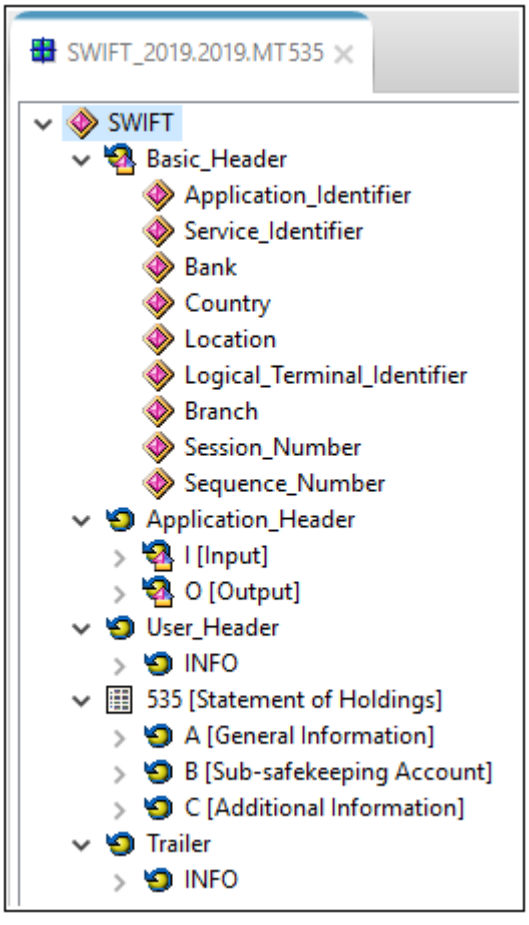


The Build Working Set dialog opens and displays a series of messages, as shown in the following image.



5. Click *Yes* to confirm your changes.

Your edited Ebix in the iIT dictionary editor should now resemble the following image.

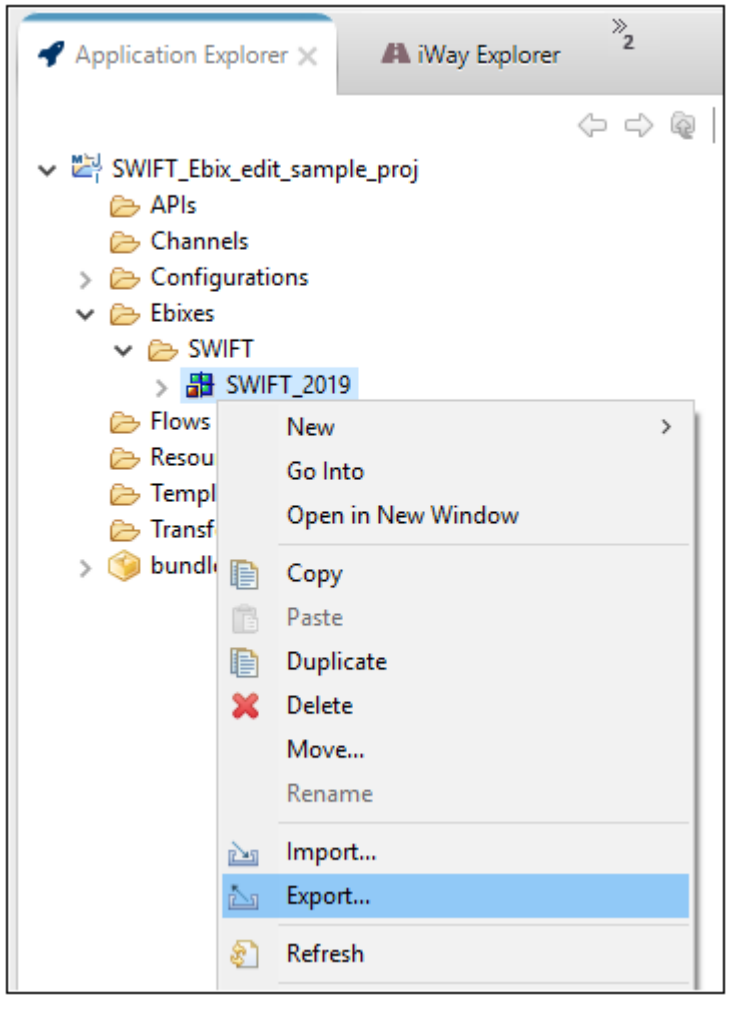


Note: The asterisk (*) character next to the file name in the tab will disappear once the edited Ebix has been saved successfully.

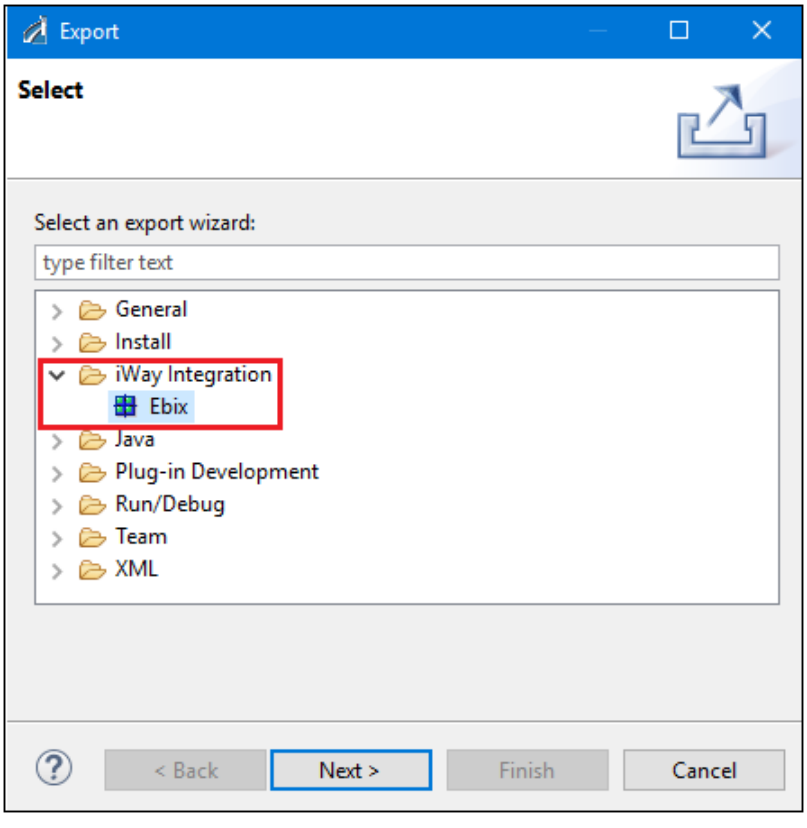
Procedure: How to Export an Ebix

To export an Ebix:

1. Right-click the *SWIFT_2019* Ebix from the Ebixes subfolder of your application project, and select *Export* from the context menu, as shown in the following image.

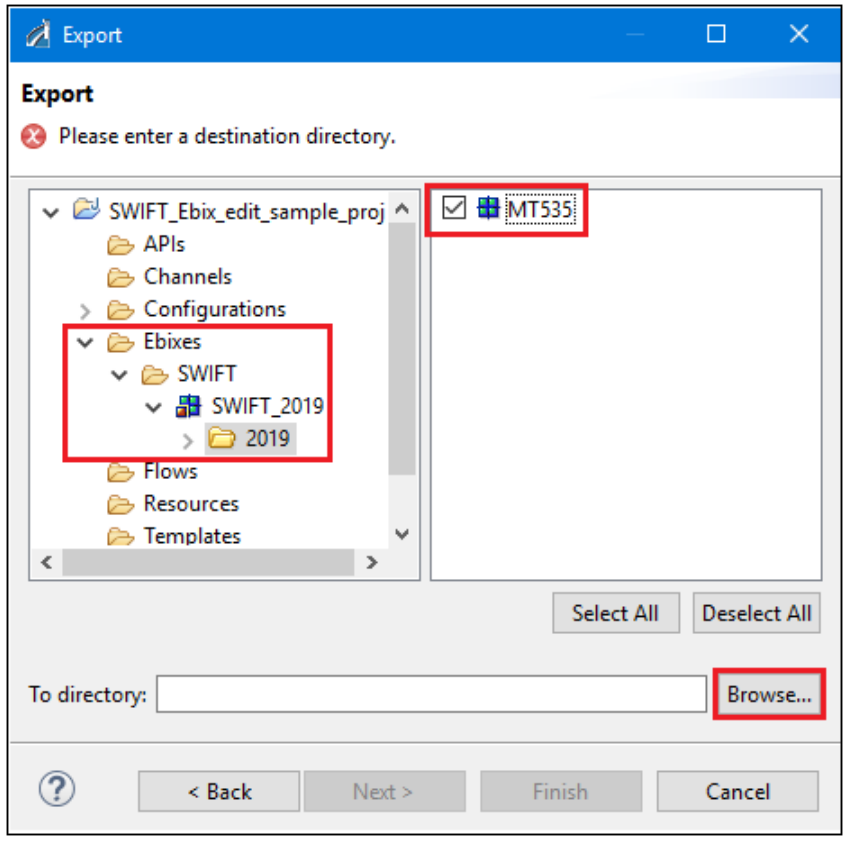


The Export dialog opens, as shown in the following image.



2. Expand the *iWay Integration* folder, select *Ebix*, and then click *Next*.

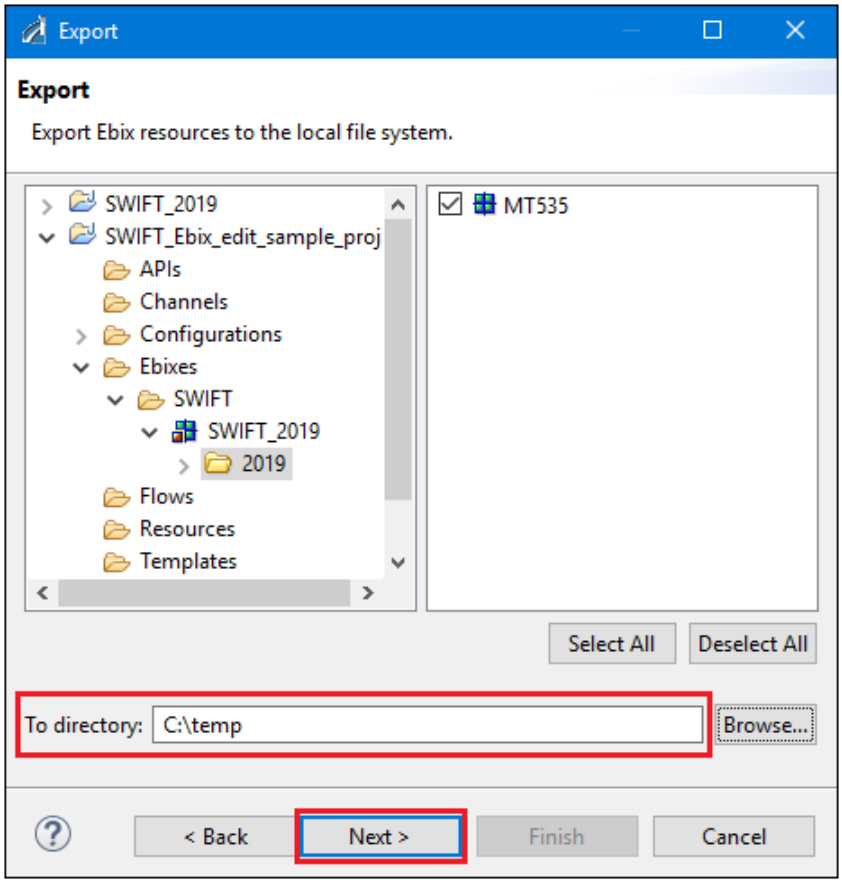
The Export pane opens, as shown in the following image.



3. Expand your application project (for example, *SWIFT_Ebix_edit_sample_proj*), *Ebixes*, *SWIFT*, *SWIFT_2019*, and then select the *2019* folder in the left pane.
4. Select the *MT535* check box in the right pane, and then click *Browse*.

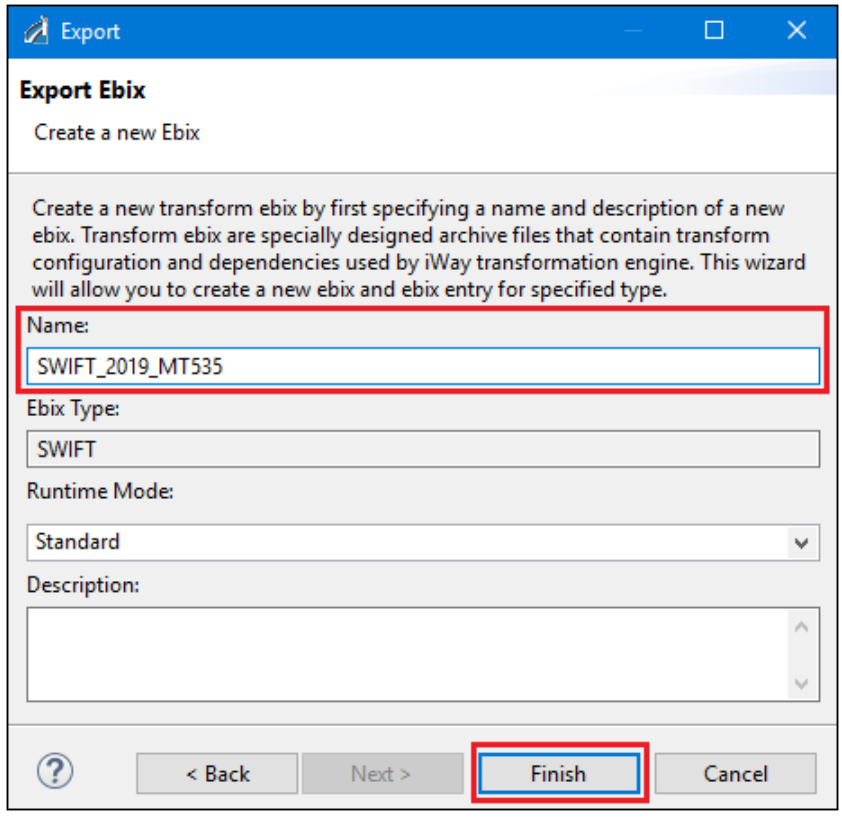
   The Export Ebix to Directory dialog opens.
5. Choose a location on your file system to store the exported Ebix, and then click *OK*.

You are returned to the Export pane, as shown in the following image.



6. Click *Next*.

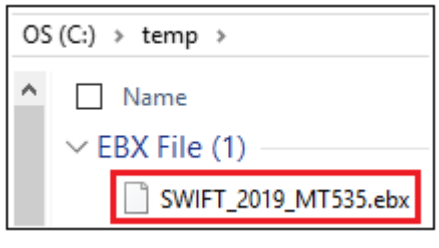
The Export Ebix pane opens, as shown in the following image.



7. Provide a name that will be used for the exported Ebix file in the Name field, select *Standard* from the Runtime Mode drop-down list, add a description (optional), and then click *Finish*.

   Your exported Ebix file is now available in the specified location on your file system.

   For example:



Appendix C

Sample SWIFT Documents

This appendix includes the following SWIFT sample documents:

- MT950 (Statement of Cash)
- MT535 (Statement of Holding)
- MT541 (Receive Against Payment)

In this appendix:

SWIFT MT950 (Statement of Cash) Sample Document

For more information on obtaining the SWIFT MT950 sample file (MT950.swift) for testing purposes, see *Extracting SWIFT User Directories and Data Samples* on page 42.

SWIFT MT535 (Statement of Holding) Sample Document

For more information on obtaining the SWIFT MT535 sample file (MT535.swift) for testing purposes, see *Extracting SWIFT User Directories and Data Samples* on page 42.

SWIFT MT541 (Receive Against Payment) Sample Document

For more information on obtaining the SWIFT MT541 sample file (MT541.swift) for testing purposes, see *Extracting SWIFT User Directories and Data Samples* on page 42.

Feedback

Customer success is our top priority. Connect with us today!

Information Builders Technical Content Management team is comprised of many talented individuals who work together to design and deliver quality technical documentation products. Your feedback supports our ongoing efforts!

You can also preview new innovations to get an early look at new content products and services. Your participation helps us create great experiences for every customer.

To send us feedback or make a connection, contact Sarah Buccellato, Technical Editor, Technical Content Management at *Sarah_Buccellato@ibi.com.*

To request permission to repurpose copyrighted material, please contact Frances Gambino, Vice President, Technical Content Management at *Frances_Gambino@ibi.com.*

Information Builders

iWay

iWay Integration Solution for SWIFT 2019 User's Guide

Version 8.0 and Higher

DN3502155.0719

Information Builders, Inc.
Two Penn Plaza
New York, NY 10121-2898